

Model Driven Architecture with Java

Gregory Cranz
Solutions Architect
Arrow Electronics, Inc.

V20061005.1351

Who am I?

- ❖ Solutions Architect
- ❖ Software Developer
- ❖ Java Early Adopter

What is it?

- ❖ **MDA – Model Driven Architecture**

- ❖ **Philosophy, Discipline & Methodology**
 1. **Platform Independent Definition**
 2. **Platform Specific Definition**
 3. **Implementation**

- ❖ **Separation of Design & Architecture**
 - **Design is Functionality**
 - **Architecture is Infrastructure**

<http://www.omg.org/docs/omg/03-06-01.pdf>

Enter Ruby on Rails



- ❖ New Language – Ruby
- ❖ New Framework – Rails
- ❖ Massive Productivity Gains

<http://www.rubyonrails.org>

Strategy

The Problem: We need to stop repeating ourselves

CRUDS

- Create
- Retrieve
- Update
- Delete
- Search

Strategy

The Problem: We need to stop repeating ourselves

1. Create `Person` class.
2. Create `PersonDAO` class.
3. Create `Person` table in database.
4. Define `PersonDAO` in Spring application context XML file.
5. Create `Person` page or action class.
6. Add `Person` pages to web framework XML configuration files.
7. Create `personList` page to list `Person` instances.
8. Create `personEdit` page to edit `Person` instances.

Strategy

The Problem: We need to stop repeating ourselves

The Answer: Define the model once & let the rest be generated for you.

1. Create `Person` Class

The Details:

Use Model-Driven Architecture

Apply Open Source Solutions

Ruthlessly Automate

MDA Solutions

- ❖ Ruby on Rails
- ❖ Oracle Application Express
(formerly HTMLDB)
- ❖ Trails
- ❖ Grails
- ❖ Compuware OptimalJ
- ❖ Etc...



TRAILS Framework:



❖ Model Driven Architecture

❖ Leverage Open Source



❖ Ruthless Automation

<http://www.trailsframework.org>

Requirements

- ❖ Java 1.5
- ❖ Apache Tomcat 5.5
- ❖ Ant 1.6
- ❖ Spring
- ❖ Hibernate
- ❖ Tapestry

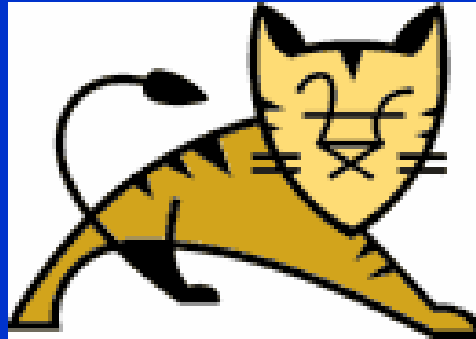
Java 1.5:



❖ JSR-220 Annotations

<http://java.sun.com>

Apache Tomcat 5.5:



- ❖ Servlet Container
- ❖ Reference Implementation
- ❖ Could be replaced with any application server

<http://tomcat.apache.org/>

Ant 1.6:



- ❖ Cross-platform Build Tool
- ❖ Native Java Tool
- ❖ Extensible
- ❖ Perfect for Automation of repetitive tasks

<http://ant.apache.org>

Spring Framework:



- ❖ *Dependency Injection / Inversion of Control* framework, designed to make configuring and connecting numerous java objects easier.
- ❖ `SessionFactory` - a Hibernate session factory
- ❖ `transactionManager` - a hibernate transaction manager
- ❖ `persistenceService` - a Trails object that provides the connection between Trails and Hibernate
- ❖ `descriptorService` - a Trails object that gathers a list of all hibernate-persisted types and decorators that add or modify meta information.
- ❖ `editorService` - a Trails object that provides an appropriate editor for a property, based on its meta information.

<http://www.springframework.org/>

Hibernate Framework:



- ❖ **OR/M & Data Persistence tool**
- ❖ **It persists POJOs: the POJOs themselves do not contain any 'obtrusive' code specific to persistence.**
- ❖ **A mapping of POJOs to database tables is done with annotations or separate xml configuration, depending on your preference.**
- ❖ **Hibernate can optionally create the database tables for you based on your mapping, if the tables do not exist yet.**
- ❖ **It features lazy loading of collections, so retrieving one object from persistent storage does not pull a complete object graph from storage.**

<http://www.hibernate.org>

Tapestry Framework:



- ❖ Component Based - HTML Fragments
- ❖ Tapestry Components consist of three parts:
 - The **editor components** are automatically placed where they are needed (text fields, ForEach components, submit buttons, links to other pages, etc.)
 - A **component specification**, defining and configuring components that are used in the HTML fragment and the Java class that provides the behavior for the component
 - A **Java class** that contains the component's behavior and an interface for the HTML presentation.

<http://tapestry.apache.org>

Trails Example

❖ ant create-project

- We are prompted for Base Directory & Project Name
- “recipe”

```
C:\Documents and Settings\aispeb\Desktop\JavaLib\trails>ant create-project
Buildfile: build.xml

check-for-apt:

create-project:
  [input] Base directory:
c:\trailsdemo
  [input] Project name:
recipe

jar-if-necessary:

update-project:
  [mkdir] Created dir: C:\trailsdemo\recipe
  [copy] Copying 824 files to C:\trailsdemo\recipe
  [copy] Copying 4 files to C:\trailsdemo\recipe
  [copy] Copying 1 file to C:\trailsdemo\recipe\context\WEB-INF
  [mkdir] Created dir: C:\trailsdemo\recipe\lib
  [copy] Copying 56 files to C:\trailsdemo\recipe\lib
  [copy] Copying 1 file to C:\trailsdemo\recipe\lib

BUILD SUCCESSFUL
Total time: 42 seconds
C:\Documents and Settings\aispeb\Desktop\JavaLib\trails>
```

Trails Example

❖ This creates our project structure:

```
<basedir>/recipe/  
<basedir>/recipe/src  
<basedir>/recipe/context  
<basedir>/recipe/context/WEB-INF  
<basedir>/recipe/lib
```

```
C:\trailsdemo\recipe>dir  
Volume in drive C has no label.  
  
Directory of C:\trailsdemo\recipe  
  
10/05/2006  09:07 AM    <DIR>          .  
10/05/2006  09:07 AM    <DIR>          ..  
10/05/2006  09:07 AM                124 build.properties  
10/05/2006  09:07 AM            7,277 build.xml  
10/05/2006  09:07 AM    <DIR>          context  
10/05/2006  09:07 AM            450 jetty-config.xml  
10/05/2006  09:07 AM    <DIR>          lib  
10/05/2006  09:07 AM    <DIR>          src  
                3 File(s)          7,851 bytes  
                5 Dir(s) 103,428,669,440 bytes free  
  
C:\trailsdemo\recipe>
```

Trails Example

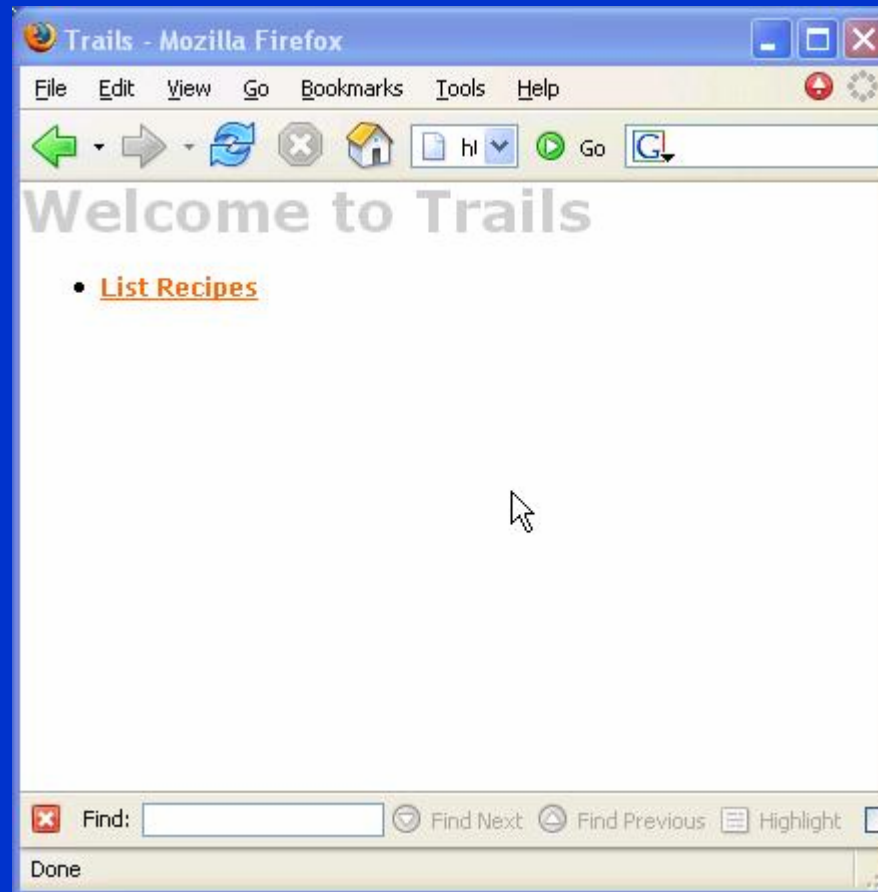
Note: @Entity annotation

Note: @Id annotation

```
package org.trails.recipe;
import java.util.Date;
import javax.persistence.Entity;
import javax.persistence.GenerationType;
import javax.persistence.Id;
@Entity
public class Recipe {
    private Integer id;
    private String title;
    private String description;
    private Date date;
    @Id(generate=GeneratorType.AUTO)
    public Integer getId() {
        return id;
    }
    public void setId(Integer id) {
        this.id = id;
    }
    public String getTitle() {
        return title;
    }
    public void setTitle(String title) {
        this.title = title;
    }
    public String getDescription() {
        return description;
    }
    public void setDescription(String description) {
        this.description = description;
    }
    public Date getDate() {
        return date;
    }
    public void setDate(Date date) {
        this.date = date;
    }
}
```

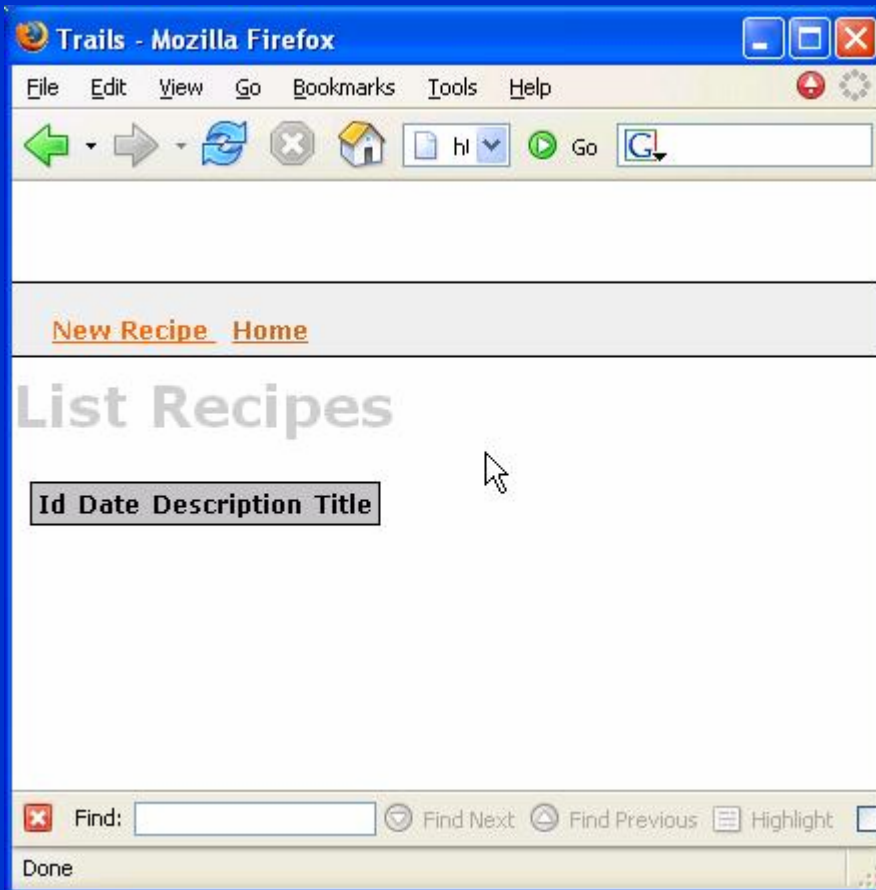
Trails Example

- ❖ ant deploy
- ❖ Step One ... and We're Done!

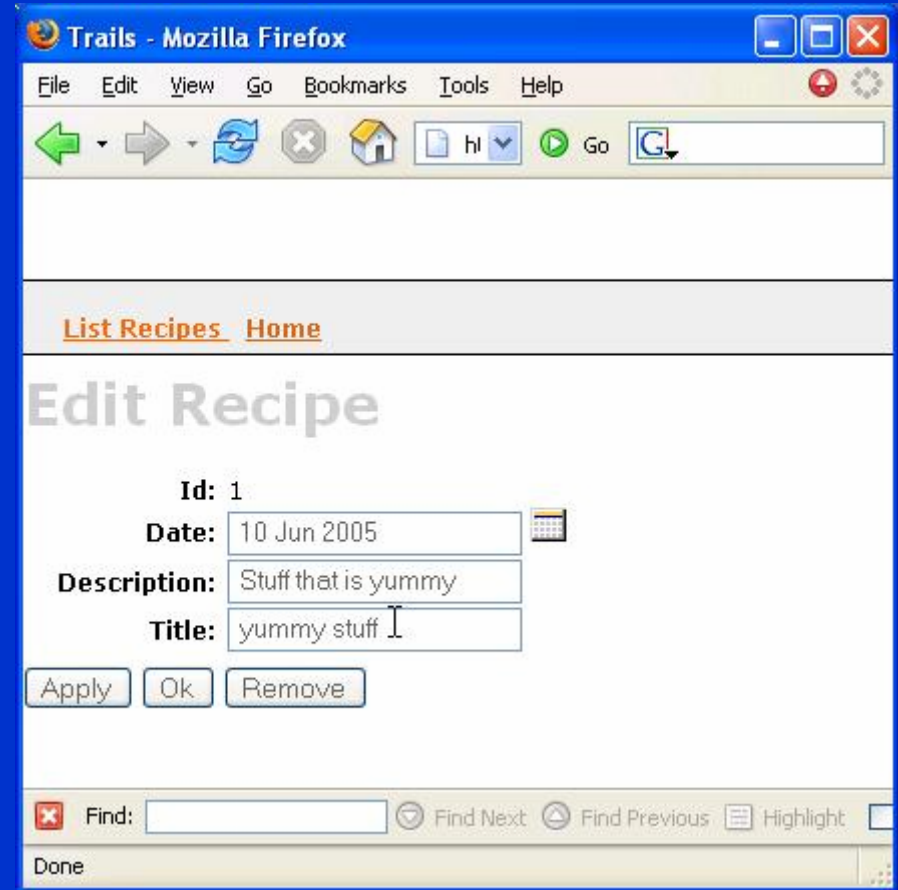


Trails Example

List View (empty)



Add a New Recipe



Trails Example

❖ Relationships

```
package org.trails.recipe;

import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.Id;
import org.apache.commons.lang.builder.EqualsBuilder;

@Entity
public class Category {
    private Integer id;

    private String name;
    @Id(generator=GeneratorType.AUTO)
    public Integer getId() { return id; }

    /** * @param id The id to set. */
    public void setId(Integer id) {
        this.id = id; }

    public String getName() {
        return name; }

    /** * @param name The name to set. */
    public void setName(String name) {
        this.name = name; }

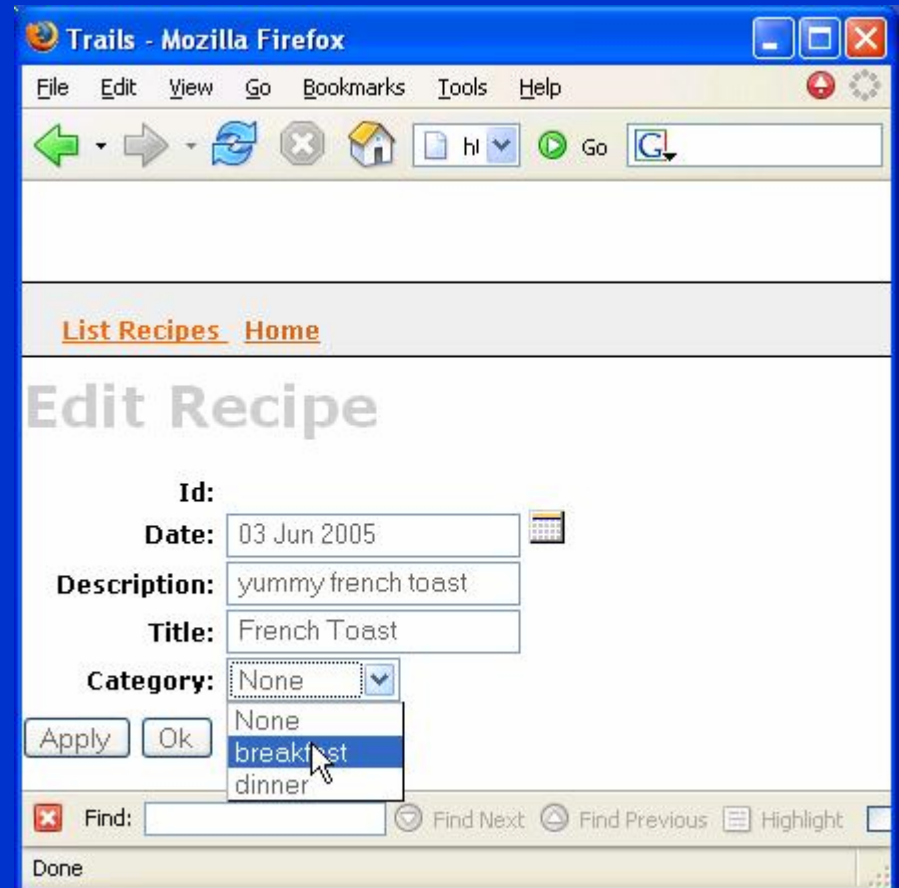
    public boolean equals(Object obj) {
        return EqualsBuilder.reflectionEquals(this, obj); }

    public String toString() {
        return getName(); }
}
```

Trails Example

❖ Relationships : many-to-one (added to Recipe)

```
private Category category;  
  
@ManyToOne  
public Category getCategory()  
{  
    return category;  
}  
  
public void setCategory(Category category) {  
    this.category = category;  
}
```



Trails Example

❖ More On Relationships

```
package org.trails.recipe;
import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.Id;
import org.apache.commons.lang.builder.EqualsBuilder;

@Entity
public class Ingredient {
    private Integer id;
    private String amount;
    private String name;

    public String getAmount() {
        return amount; }
    public void setAmount(String amount) {
        this.amount = amount; }

    @Id(generator=GeneratorType.AUTO)
    public Integer getId() {
        return id; }

    public void setId(Integer id) {
        this.id = id; }
    public String getName() {
        return name; }
    public void setName(String name) {
        this.name = name; }
    public boolean equals(Object obj) {
        return EqualsBuilder.reflectionEquals(this, obj); }
    public String toString() {
        return getAmount() + " " + getName(); }
}
```


Trails Example

❖ More on Relationships : one-to-many

```
private Set<Ingredient> ingredients = new HashSet<Ingredient>();

@OneToMany(cascade=CascadeType.ALL)
@JoinColumn(name="recipeId")
@Collection(child=true)
public Set<Ingredient> getIngredients()
{
    return ingredients;
}

public void setIngredients(Set<Ingredient> ingredients)
{
    this.ingredients = ingredients;
}
```

Trails Example

❖ Relationships : One-to-Many

The image displays two side-by-side screenshots of a web application running in Mozilla Firefox. Both windows show the URL `http://localhost:8080/app`.

The left window, titled "Trails - Mozilla Firefox", shows the "Edit Recipe" page. It features a navigation bar with "List Recipes" and "Home" links. The main content area includes a form with the following fields and controls:

- Category: (dropdown)
- Date: (calendar icon)
- Description:
- Id:
- Ingredients:
- Title:
- Buttons:

The right window, also titled "Trails - Mozilla Firefox", shows the "Add Ingredient" page. It features a navigation bar with "List Ingredients" and "Home" links. The main content area includes a form with the following fields and controls:

- Id:
- Amount:
- Name:
- Buttons:

Trails Example

❖ Relationships : One-to-Many

Trails - Mozilla Firefox

File Edit View Go Bookmarks Tools Help

http://localhost:8080/app

[List Recipes](#) [Home](#)

Edit Recipe

Category

Date

Description

Id 1

Ingredients **1/3 cup lettuce**

Title

Trails - Mozilla Firefox

File Edit View Go Bookmarks Tools Help

http://localhost:8080/app

[List Recipes](#) [Home](#)

Edit Recipe

Id 1

Title

Description

First Cooked On

Ingredients **1 tsp foobar dressing**

1 cup lettuce

Category

Thank You!

**Contact:
Gregory Cranz
Solutions Architect
Enterprise Architecture
gcranz@arrow.com**