Cairngorm for ColdFusion Developers

RIA: Flex / AJAX / AIR Track

Brian Rinaldi

www.remotesynthesis.com

CF.Objective()

About This Presentation

- Intended Audience:
  - For beginning Flex developers and/or those with minimal exposure to the Cairngorm framework
  - Not intended to debate the merits of Cairngorm
- Organization:
  - This discussion is organized according to a standard Cairngorm event flow (your actual coding order may vary)

Who am I?

- Adobe Community Expert
- Flex Champion
- Boston ColdFusion User Group Manager
- Organizer of Flex Camp Boston (Dec. 2007)
- Creator of the Illudium PU-36 Code Generator open-source project
- Frequent author to publications including Fusion Authority Quarterly Update, Adobe Edge, Sitepoint, ColdFusion Weekly Podcast
- Blog - RemoteSynthesis.com
  - ColdFusion Open-Source List
  - Weekly ColdFusion Open-Source Update

Why Does Cairngorm Seem So Difficult?

The "Duck Dodgers" Approach - tends to over-explain the multitude of patterns within Cairngorm

The "Porky Pig" Approach - deal with Cairngorm as an easily-repeatable, iterative process

* See movie clip for context. (-4:39)

This presentation will attempt to be different through:

- Focusing on the how rather than the why
- Boiling Cairngorm development down to a step-by-step process
- Relating complex concepts/practices to their ColdFusion equivalent (where one exists)

Based upon my article in the Fusion Authority Quarterly Update (Volume 2 Issue 2)

Why Should You Care?

By providing this consistent architecture and proven design patterns, using Cairngorm can offer a number of benefits to your Flex application, including:

- Centralized model (i.e. the Model Locator)
- A centralized event/response system
- Separation of concerns along the lines of the Model-View-Controller pattern
  - isolate business logic
  - simplify maintenance
- Straightforward and documented structure for application building within a team environment
- A well-known approach towards Flex development that a wide-range of Flex developers are already familiar with
  - If you develop Flex, you will encounter Cairngorm at some point

Getting Started - Installing Cairngorm

- Download - http://labs.adobe.com/wiki/index.php/Cairngorm
  - Current version is 2.2.1
- Place the Cairngorm.swc anywhere on your computer
- Create your Flex/AIR project in Flex Builder
- Under “Flex Build Path"
  - Choose the "Library Path" tab
  - Click on the "Add SWC" button
  - Locate your Cairngorm.swc and click “ok”

Basic Terms and Concepts

1. Control (FrontController):
    - Determines where to route the request (Command class).
    - Same concept as ColdFusion frameworks such as Mach ii, Model-Glue or Fusebox
2. Command
    - Executes the service call via a Delegate, returning the data to result() (success) or a fault() (fail)
3. Delegate
    - Encapsulates access to a business service.
4. Model (ModelLocator)
    - State of the application and its variables (conceptually similar to application scope in ColdFusion)
    - Using data-binding any component in the application that is bound to the model is immediately updated when a variable within the model is updated

When to Use Cairngorm

- You have a decent level of comfort with Flex
- Your project is of moderate to high level complexity
- Your project has several or more "use cases"
- You are working with a group of developers (or other developers will maintain the code upon completion)

i.e. Most Flex applications would benefit from using a structured approach like Cairngorm

Further Reading

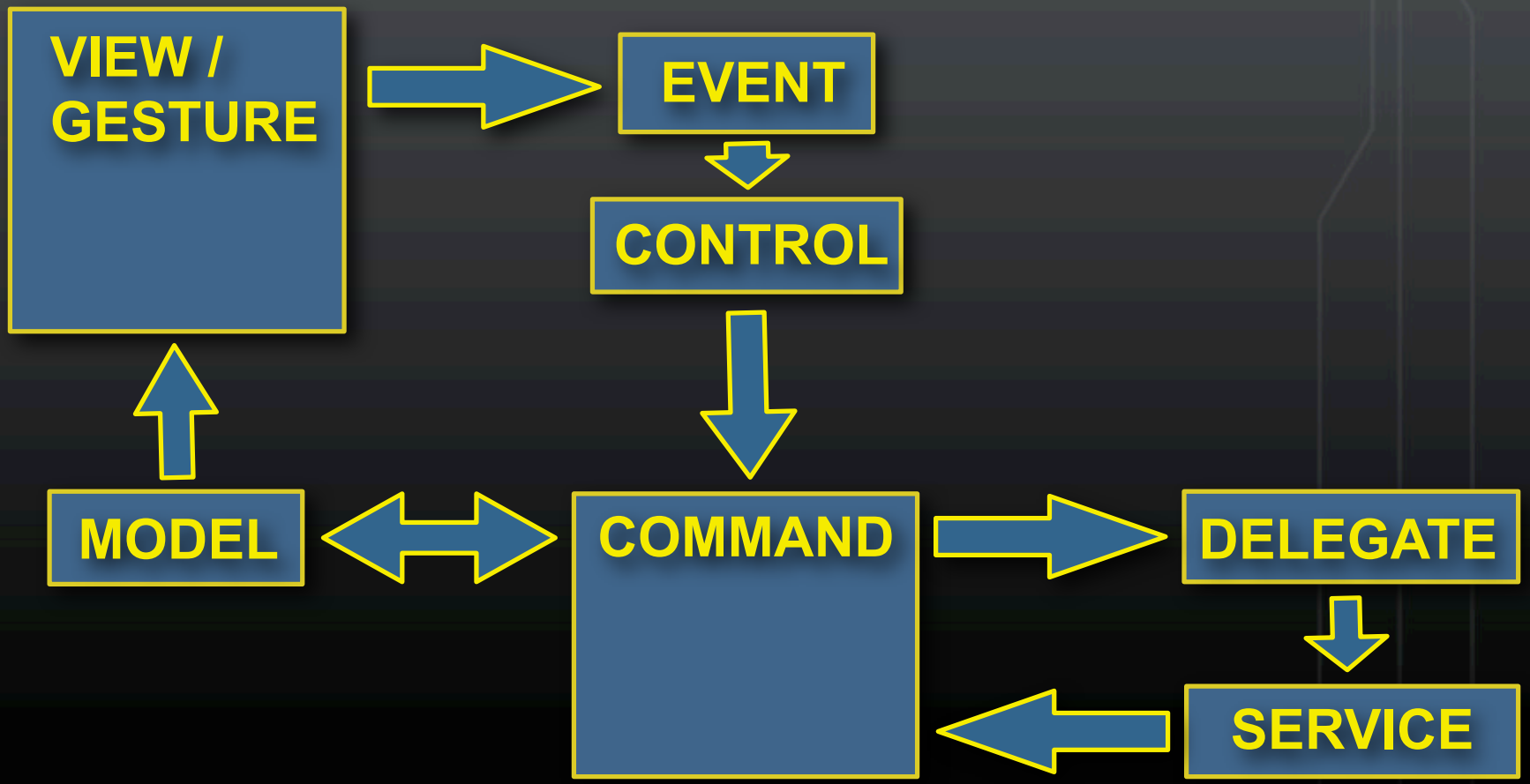
http://weblogs.macromedia.com/swebster/archives/2006/08/why_i_think_you.cfm

http://www.leifwells.com/index.cfm/2007/1/8/When-is-Cairngorm-too-much-for-a-Flex-2-application

For a different approach see: http://www.adobe.com/devnet/flex/articles/blueprint.html

Cairngorm Event Process

Cairngorm events are triggered by user interaction or application events



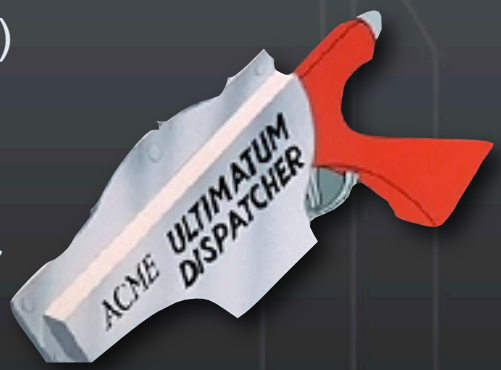
Step 1a - Architecting the View

Cairngorm doesn't care how you architect your view. However, some practices are *recommended*:

- Separate reusable portions of your application into components to improve maintainability
- Utilize data-binding to auto-update model changes
- Define constants for view-state management

Step 1b - Announce an event

- User interaction or application process occurs (gesture)
- An internal function is called
- Event is initialized
- Event is announced via the CairngormEventDispatcher
  - In package com.adobe.cairngorm.control
  - Singleton class responsible for broadcasting events to the FrontController



Syntax:

```
CairngormEventDispatcher.getInstance()
  .dispatchEvent( eventObject );
```

Step 2 - Creating an Event

- Simple class that has a contructor and properties (if necessary)
- Every event extends com.adobe.cairngorm.control.CairngormEvent
- Can contain data to be passed along with the event type
  - super(EVENT_FOO);
- Must establish the event type with the front controller
- Must implement clone() method
  - see http://life.neophi.com/danielr/2006/06/events_in_flex_2.html

Step 3 - Modify the Controller

Establish the Command class that will respond when this event is announced. The Cairngorm framework makes sure the proper call is made.

Syntax:

```
addCommand(FooEvent.EVENT_ANNOUNCE_FOO,FooCommand);
```

Step 4 - Creating a Command

- Taken from "Gang of Four" pattern catalog - allows you to invoke a command without requiring knowledge of what the command does.
- Implements the com.adobe.cairngorm.commands.ICommand interface
  - Command interface deprecated as of 2.1
- Must have an execute() method that accepts a CairngormEvent
- Often has two additional methods (when connecting to remote data)
  - Successful result handler
  - Failed result handler
- Most frequently, the command will announce an external data call via the delegate
- Can modify the model
- Does not negate the need to create your own "business objects" to abstract reusable portions of business logic

Step 5 - Creating a Delegate

Note: Steps 5 and 6 are only necessary for events that require access to remote data.

- Abstracts the specifics of the server side implementation
- Allows remote calls to be reusable
- Does not modify the model
- Makes the appropriate asynchronous remote service call
  - Remote calls are centralized via the ServiceLocator class
- I generally align each delegate with a ColdFusion service (remote facade) matching the method names

Step 6 - Call Service

Services are contained within an ServiceLocator MXML file

- Included in your root MXML file
- Can define web services, HTTP requests or Remote Object calls
  - For ColdFusion you will most frequently use RemoteObject calls to the predefined "ColdFusion" destination

Step 7 - Creating a Value Object

Value Objects are not a requirement of Cairngorm but you will generally want to use them if you wish to take advantage of the type translation between ColdFusion and Flex (specifically CFC types). They also give “meaning” to your data - for example, instead of having FirstName and LastName, you have a User.

The simplest way to create a VO based upon a CFC is to use the built in generator included in the ColdFusion Extensions for Eclipse.

- Right-click CFC
- ColdFusion Wizards > Generate AS Class (based on CFC)

Your AS class properties will match the properties defined within the <cfproperty> tags in your CFC

Rinse and Repeat

Essentially you will find that you repeat this process while building each event. After you gain some experience you may find that you don't follow the steps in this particular order.

Additional Resources

Cairngorm Class Documentation

- http://www.cairngormdocs.org/docs/cairngorm_2_1/

Cairngorm Docs

- http://cairngormdocs.org/

Cairngorm Interactive Diagram

- http://www.cairngormdocs.org/tools/CairngormDiagramExplorer.swf

Cairngorm Tutorial by Stephen Webster of Adobe Consulting

- http://www.adobe.com/devnet/flex/articles/cairngorm_pt1.html

Cairngorm Tutorial By David Tucker

- http://www.davidtucker.net/category/cairngorm/

Cairngorm Tutorial by Jeffry Houser

- http://www.jeffryhouser.com/index.cfm/2007/2/15/How-do-you-learn-Cairngorm

FIN

Questions?