Architecting AIR 1.0 Applications with ActionScript 3

By Clint Modien – Esria Inc. Feb, 27th, 2008

About Me – Clint Modien

- Technical Architect for Esria Inc.
  - Esria is a software consulting firm based out of San Jose, CA
    - Flash, Flex, AIR, .NET, Java, SQL
    - Esria is Hiring ☺
- Doing dev since the dot bomb in 2000 doing .NET, Java, SQL, Web and Desktop apps
  - Flex since Feb of 04'
- Companies I've done ActionScript work for:
  - IBM, Yahoo, Kodak, Adobe, Fidelity, Morgan Stanley, Ribbit
- Author for Wrox's Professional AIR book due out in the Second Half of 08'
  - ***This presentation is an overview of one of the chapters in the book.***

What is Architecture?

- Software Architecture is about the “Big Rocks” in a project.
  - Analogy – Big Considerations
- Architecting to Scale
  - Dog House - Typical Flash Project (1 – 2 Developers)
  - Home – Typical Flex Project (2 – 8 Developers)
  - Office Building – Nothing Typical About This (9 - Hundreds of Developers)
- Costs of Change
  - As a definition
- Ilitiy Requirements
  - Maintainability, Reusability, Performance

Overview

- AIR's Architectural Big Rocks
- Software Development Patterns for AIR
- Micro-Architectures
- Automated Builds
- Unit Testing
- Data Services Options
- Advanced Project Setup
- Modeling and Documenting
- Common Problems
- AIR Limitations
- Extending AIR through 3rd Party Tools
- AIR Alternatives

AIR's Architectural Big Rocks

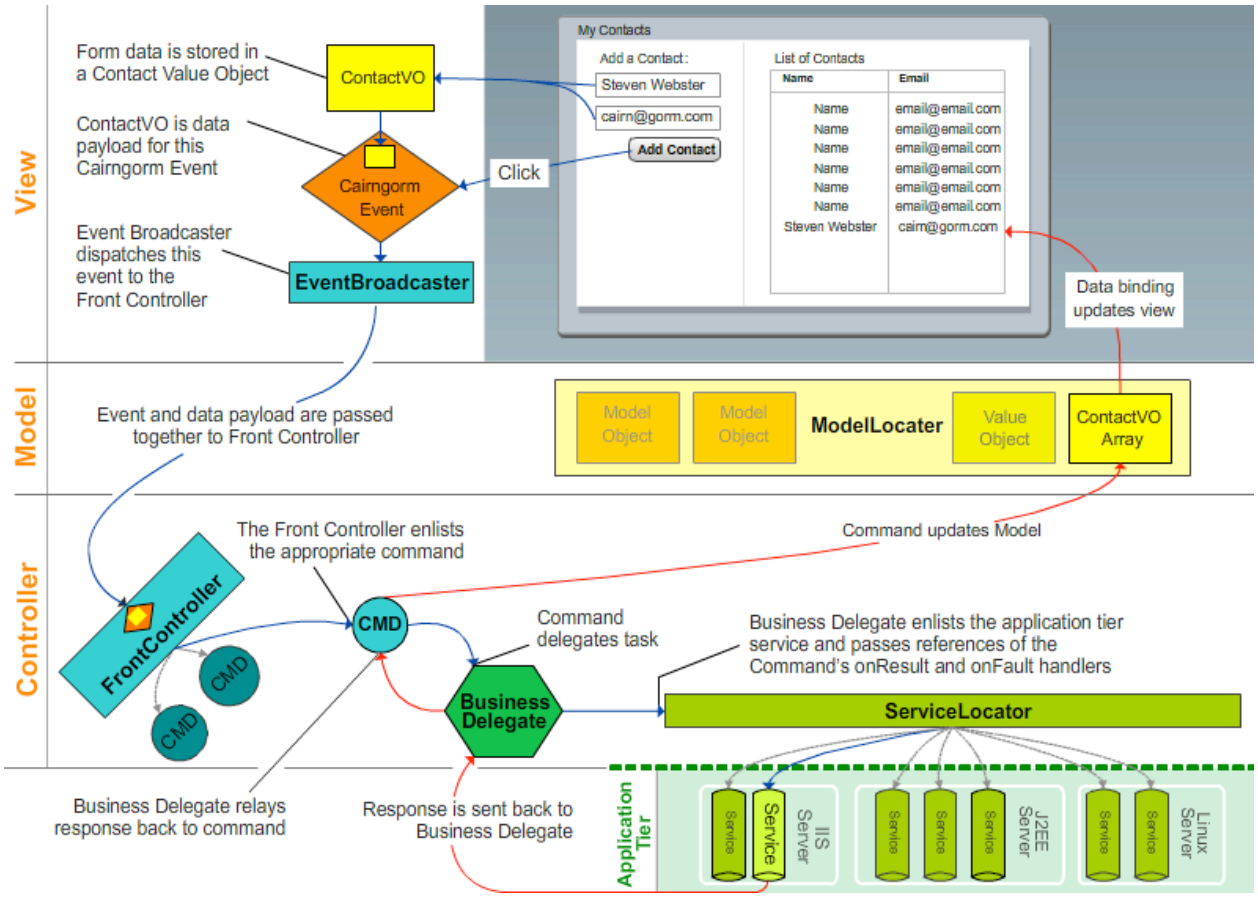
- Cross Operating System Compatibility
  - File System, Menus, Dock and Tray Icons, Alerts
- Offline / Online Occasionally Connected Clients
  - Network API, SQL API, File System API
- Security
  - Code Signing, Sandboxes
- JavaScript Considerations
  - Top Level HTML Application, Reuse exiting HTML in ActionScript, Script Bridge
- Sharing Code Between a Web and Desktop Deployment
- Application Update API

Architectural Patterns for AIR and AS3

- Class Object Model and MVC
- Object Relational Mapping
  - Esria's VOFactory
  - ORM Mapping Layer for Local SQL API
- Service Facades and Value Objects
- Session Patterns (Database, Middle-Tier, Client-Side)

MVC Micro Architectures - Cairngorm

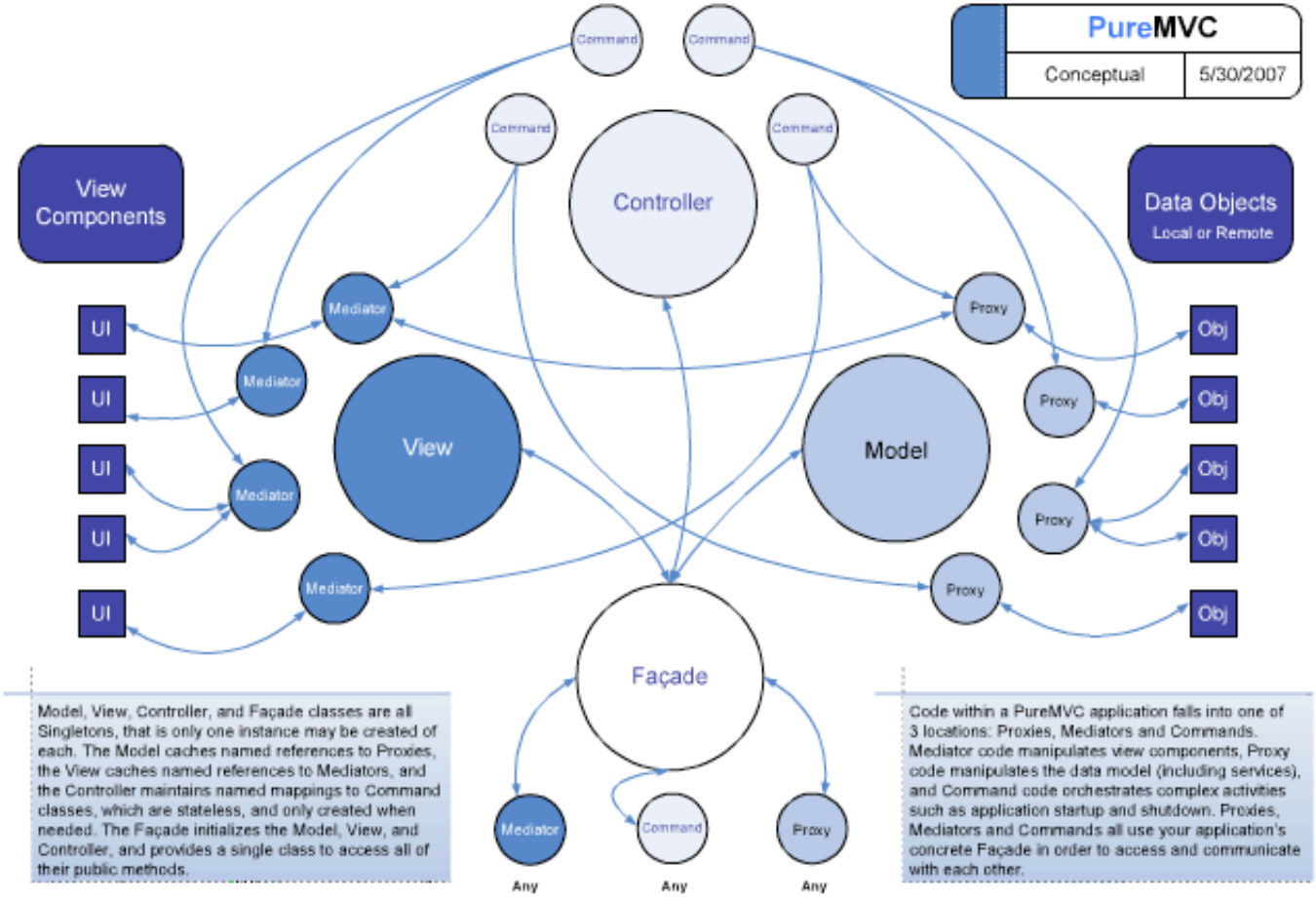
- Cairngorm



cairngormdocs.org

MVC Micro-Architectures - PureMVC

- Pure MVC



puremvc.org

Automated Builds

- This almost doesn't need to be said but if you're running more than a one man show you should have a build setup that checks out code from your code repo (SVN, Perforce, TFS) and builds it.
- Some technologies are (Cruise Control, Cruise Control.NET) (thoughtworks.com)
- Write a build script with language like Ant, or Ruby if you use Rake to build your project from the command line
- Build your code, unit tests, and docs
- Run your tests
- Report your build success or failure via email

Modeling and Documenting

- Use Sparx System's Enterprise Architect to Model code before or after you write it.
  - UML Modeling tool
  - $200 US for the Pro Version, $240 US for the Corporate (Multi User via DB)
  - Forward & Reverse Engineering for various languages including ActionScript 3.
  - HTML Export, Compatible with Visio and anything that can talk XMI
  - http://www.sparxsystems.com.au
  - Automatable from the command line
- ASDoc
  - Write comments in your code for every function, var and class
  - Format them so you can generate documentation from them using the ASDoc tool that comes with the Flex SDK
  - Run the ASDoc tool as part of your automated build

Unit Testing

- FlexUnit / ASUnit
- Do it from the beginning of the project… Make testability an “ilitiy” requirement.
- Drastically alters the way you write code
  - As a rule of thumb at least 50% of your code base should be tests (Write as many lines of code for unit tests as for actual code.) This is actually calculable using “Cyclomatic Complexity”
  - Pragmatic Programmer’s “Right BICEP” approach tells you what to test (Hunt, Thomas 06’)
- Code Coverage Utility coming from Alex Uhlman out of Adobe Consulting
- Mocking / Faking Objects
  - Code to an interface not an implementation
  - Extend classes you cannot override
- Run your ActionScript/JavaScript unit tests while you dev
- Run them as part of your automated build process with AIRRunner (http://airrunner.org)

Data Services Options

- RTMP vs. Flex Remoting vs. Web Services vs. XML
  - XML is slow… Remoting is fast (James Ward's Blog benchmarks)
  - RTMP is connected… all others are stateless
- Remoting is for more than just Java
  - WebORB supports .NET, PHP, Ruby and Java (themidnightcoders.com)
  - WebORB supports RTMP
- There's a whole chapter on Data Services in the book

Enterprise Project Setup

- Project Directory Structure
- Component Architecture
  - Common SWC with a Unit Test app
  - Logic SWC with a Unit Test app
  - Harnesses
  - Applications
- Application Modularity
  - Cairngorm and Modules

Common Problems

- Logging / Tracing
- Exception Management
- Metrics Gathering / Reporting

AIR Limitations

- Native Code Invocation
- PDF Generation
- Run on Linux (Next Ver. In Q3 of 08')
- Remote runtime class loading
  - Must use Script Bridge

Extending AIR through 3rd Party Tools

- WebORB for .NET
- Merapi for Java (formerly Artemis)

Questions?

- Email: clintm@esria.com
- Blog http://clintm.esria.com