



- An Introduction

Aksel Gresvig, RealEyes Media



(realeyes)



The Model-View-Controller Meta-Pattern

- GUIs not only mostly required but very complex today
- MVC increasingly popular with modern applications
- Aims to separate the business logic, data, and user interface of an application
- Separation of code promotes reuse of data, visual interfaces, and logic
- Applies well to Flash/Flex due to emphasis on visuals

PureMVC Concepts

- Extends concepts of the MVC pattern to avoid any potential tight coupling
- The three tiers:
 - *The View* – made up of view components and *Mediators* that handle their logic
 - *The Model* – all data interaction is handled by *Proxies*
 - *The Controller* – application-wide logic is handled by *Commands* that communicate and interact with both *Proxies*, *Mediators*, and other *Commands*
- The *Application Façade* Singleton
 - Makes it possible for all actors to talk to each

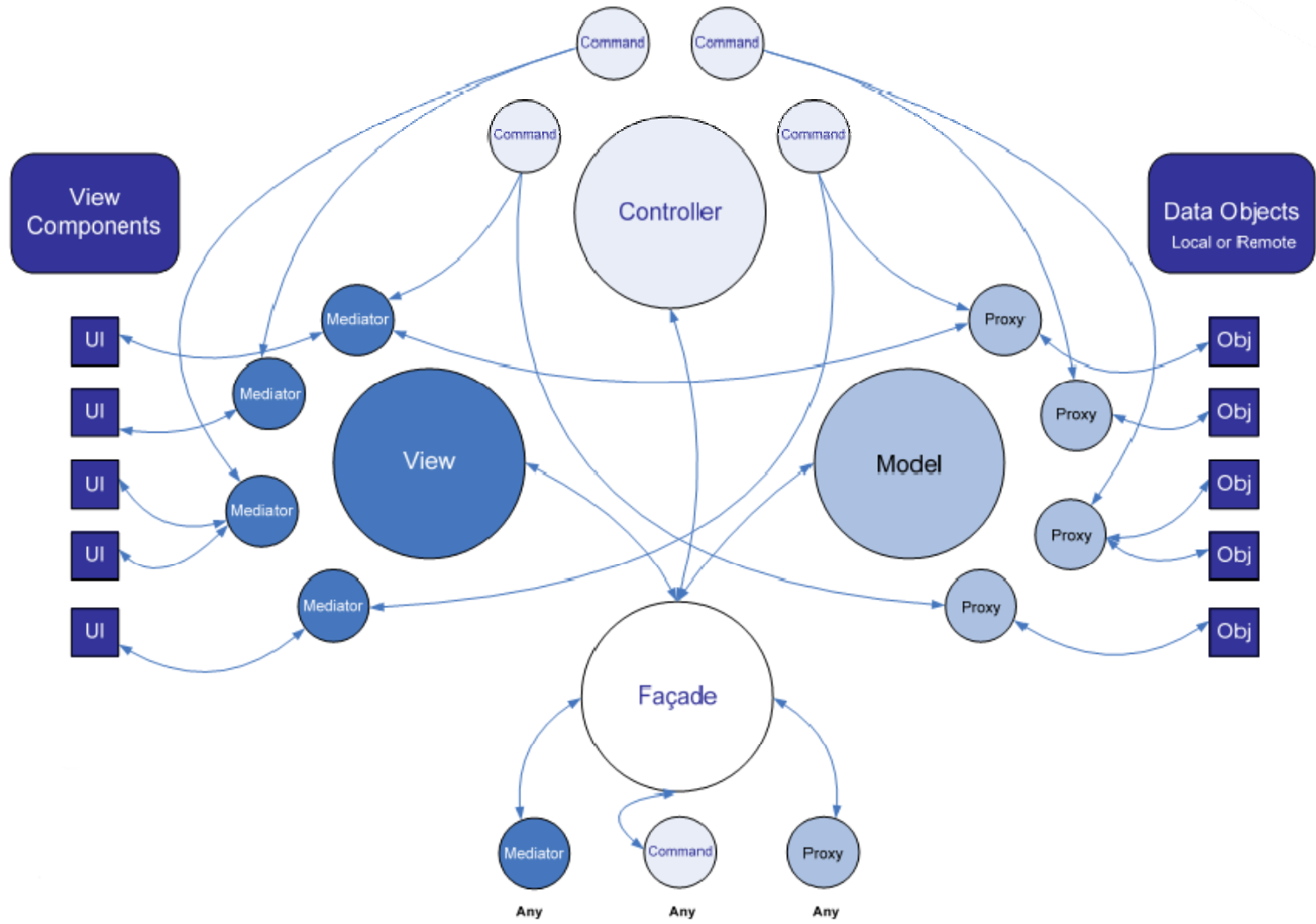
Internal communication

- Internal communication via *Notifications*
- A Notification can be sent from anywhere in the application and listened for anywhere in the application (except Proxies)
- Objects can be passed along with Notifications
- Commands are registered to execute when a particular Notification fires

Conceptual Diagram

puremvc framework

CONCEPTUAL DIAGRAM AND INTRODUCTION



Strengths & Benefits

- Achieves goal of code separation
- Reduces confusion over application layers, class roles, responsibilities and collaborations
- Platform independent
- Programmed to interfaces – hides complexity from developer

Strengths & Benefits

- Promotes reuse!
 - Makes it very easy and fast to expand an application and add functionality
 - Separation and information hiding makes it easier to maintain the application by different developers (given they understand PMVC)
- Notifications are cool
 - Allows for great communication *with* data passing
 - Easy to send, easy to receive
 - Can be received by many actors simultaneously
 - Really eases expansion
 - Well documented (<http://www.realeyes.org/>)

Weaknesses

- Complexity
 - Model, View, Controller, Mediators, Proxies, Commands, Façades, Delegates, Helpers, CFCs, VOs ++
 - A lot of files due to high separation, requires strict structure
 - Takes time and work to set up properly, can be confusing
 - A lot of interface implementation and subclassing
- Not impossible to “break the rules” of communication/information sharing
- For small apps it’s quicker to avoid separation

Links

- Official site:
www.puremvc.org
- My Intro Writeup:
<http://aksel.realeyes.com>