

SOEN 387 Web-based Enterprise Application Design

Stuart Thiel

Concordia University
Department of Computer & Software Engineering

Fall, 2015

MVC

Architectural
Styles

3-Layer Styles

Architectural
Patterns

Outline

MVC

Architectural Styles

3-Layer Styles

Architectural Patterns

MVC

Architectural
Styles

3-Layer Styles

Architectural
Patterns

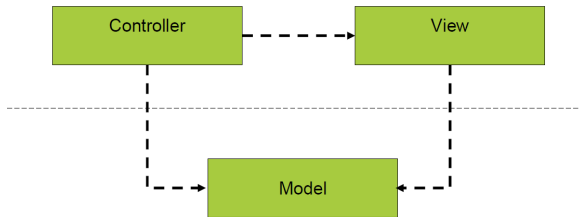
What is MVC

- ▶ Model View Controller
 - ▶ An Architectural Pattern
 - ▶ Separate internal representation from presentation
 - ▶ *Everythingtm* is MVC these days

MVC and Controllers

- ▶ Model View Controller
- ▶ Can represent overall System
- ▶ Can represent Use-Case Scenarios
- ▶ Larman's "Applying UML and Patterns"
- ▶ Who handles system events?
 - ▶ What are the system events?
- ▶ Both Controllers can have their uses

General MVC Look



- ▶ Delegate to Model for domain logic
- ▶ Pass part of model to the view

HelloWorldMVC Tutorial

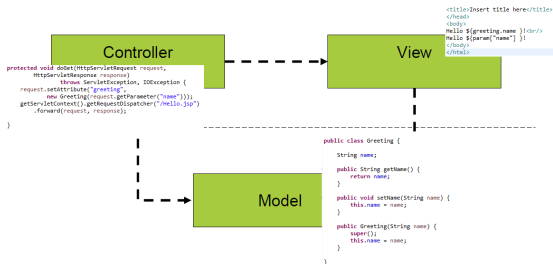
Stuart Thiel

MVC

Architectural
Styles

3-Layer Styles

Architectural
Patterns



What Are Architectural Styles?

- ▶ http://en.wikipedia.org/wiki/Software_architecture_styles_and_patterns
- ▶ "...a named collection of design decisions that are applicable in a given development context..."

Do You Know Any?

- ▶ Ever heard the term?
- ▶ Can you name any?

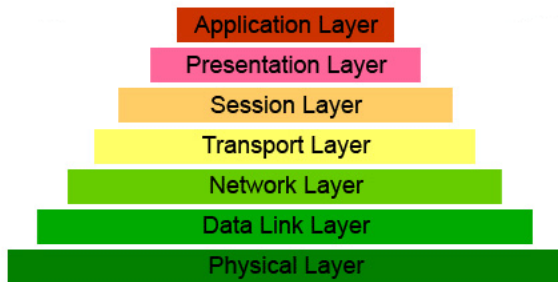
Do These Look Familiar?

- ▶ http://en.wikipedia.org/wiki/Software_architecture_styles_and_patterns
- ▶ Structural
 - ▶ Component-based
 - ▶ Layered
 - ▶ Pipe & Filter
- ▶ Messaging
 - ▶ Events
 - ▶ Publish-Subscribe
 - ▶ Asynchronous Messaging
- ▶ Distributed
 - ▶ Client-Server
 - ▶ Shared Architecture
 - ▶ Broker
 - ▶ Peer-to-Peer
 - ▶ Service Oriented

Webapps Often Use These

- ▶ Pipe Filter
 - ▶ But we won't talk about this much
- ▶ Layered

The Seven Layers of OSI

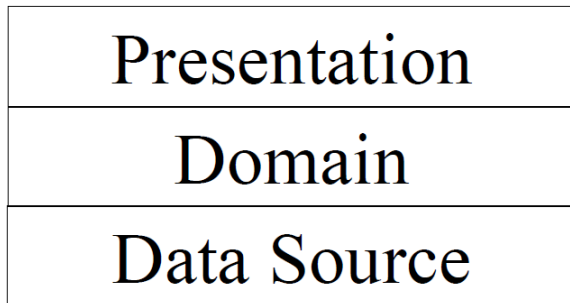


- ▶ Simplifies Design
- ▶ Isolated most interaction within layers
- ▶ Supports being portable
- ▶ Supports distribution?

Motivation for Layers

- ▶ Big mix of WEA Critical Functionality
- ▶ Lots of concurrent data access
- ▶ Lots of different views
- ▶ Lots of different business logic
- ▶ Lots of data
- ▶ How to integrate?

Fowler's Enterprise Application Layers



Larman's Enterprise Application Layers

- GUI windows
- reports
- speech interface
- HTML, XML, XSLT, JSP, Javascript, ...

Presentation
(AKA Interface, UI, View)

- handles presentation layer requests
- workflow
- session state
- window/page transitions
- consolidation/transformation of disparate data for presentation

Application
(AKA Workflow, Process, Mediation, App Controller)

- handles application layer requests
- implementation of domain rules
- domain services (*POS, Inventory*)
- services may be used by just one application, but there is also the possibility of multi-application services

Domain(s)
(AKA Business, Business Services, Model)

- very general low-level business services used in many business domains
- *CurrencyConverter*

Business Infrastructure
(AKA Low-level Business Services)

- (relatively) high-level technical services and frameworks
- *Persistence, Security*

Technical Services
(AKA Technical Infrastructure, High-level Technical Services)

- low-level technical services, utilities, and frameworks
- *data structures, threads, math, file, DB, and network I/O*

Foundation
(AKA Core Services, Base Services, Low-level Technical Services/Infrastructure)

A Quick Look At Our Cast

Stuart Thiel

MVC

Architectural
Styles

3-Layer Styles

Architectural
Patterns

