MVC

CE-4101
Especificación y diseño
de software

MVC

- Solución probada para lograr la separación entre la presentación y la lógica
- Patrón arquitectura que maneja la presentación de la información y la entrada de datos en múltiples tipos de clientes
- Los sistema que implementan el MVC tienden a ser mas flexibles y extensibles ya que facilita cambiar la presentación de la aplicación sin afectar su comportamiento

Modelo

- Maneja los datos que serán desplegados por la vista y la lógica del negocio.
- Mantiene el estado de la aplicación

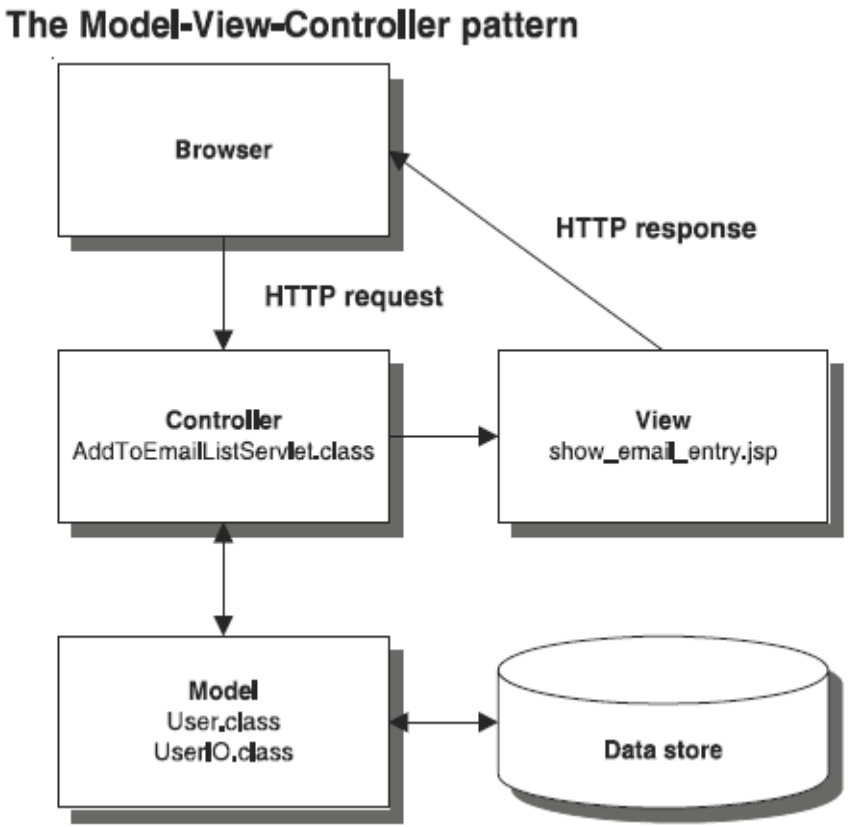
Vista

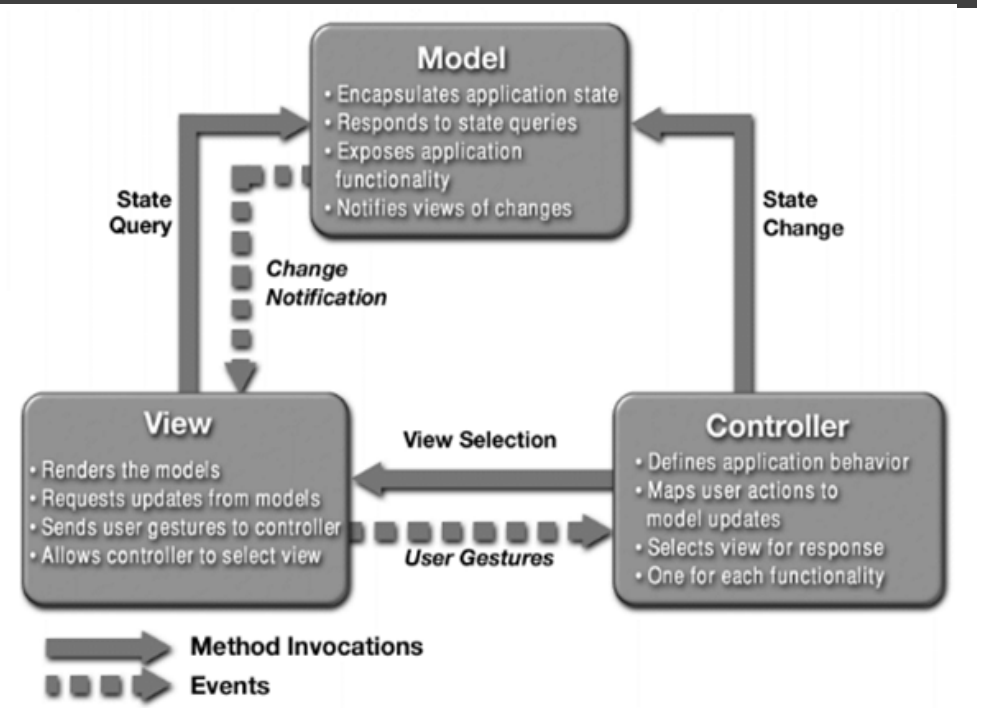
- Salida del modelo.
- Vista es determinada por el controlador
- Nunca debe implementar la lógica del negocio

Controlador

- Define el comportamiento de la aplicación
- Asocia actiones a funciones de la aplicaciones
- Determina la vista
- Examina y extrae los parámetros del request

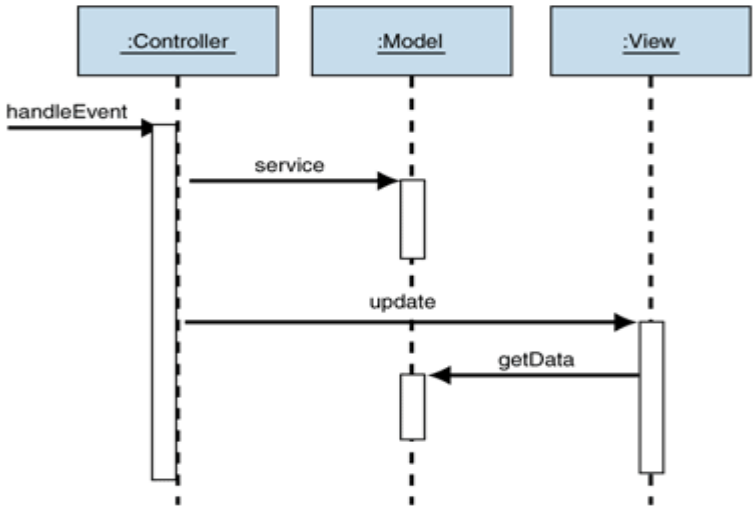
MVC


The Model-View-Controller pattern
Browser
HTTP response
HTTP request
Controller
AddToEmailListServlet.class
View
show_email_entry.jsp
Model
User.class
UserIO.class
Data store


Model
• Encapsulates application state
• Responds to state queries
• Exposes application functionality
• Notifies views of changes
State Query
Change Notification
State Change
View
• Renders the models
• Requests updates from models
• Sends user gestures to controller
• Allows controller to select view
View Selection
User Gestures
Controller
• Defines application behavior
• Maps user actions to model updates
• Selects view for response
• One for each functionality
Method Invocations
Events

MVC

- Una variación del modelo es el MVC passive model
- El modelo pasivo es empleado cuando un controlador manipula de manera exclusiva el modelo.
- El controlador modifica al modelo y luego informa a la vista que el modelo ha cambiado y por tanto debe ser refrescada
- Muy utilizado en sistema Web ya que no hay una forma simple en que el browser obtenga actualizaciones asíncronas del servidor. El browser despliega la vista y responde a los inputs del usuario pero no detecta cambios en los datos del server

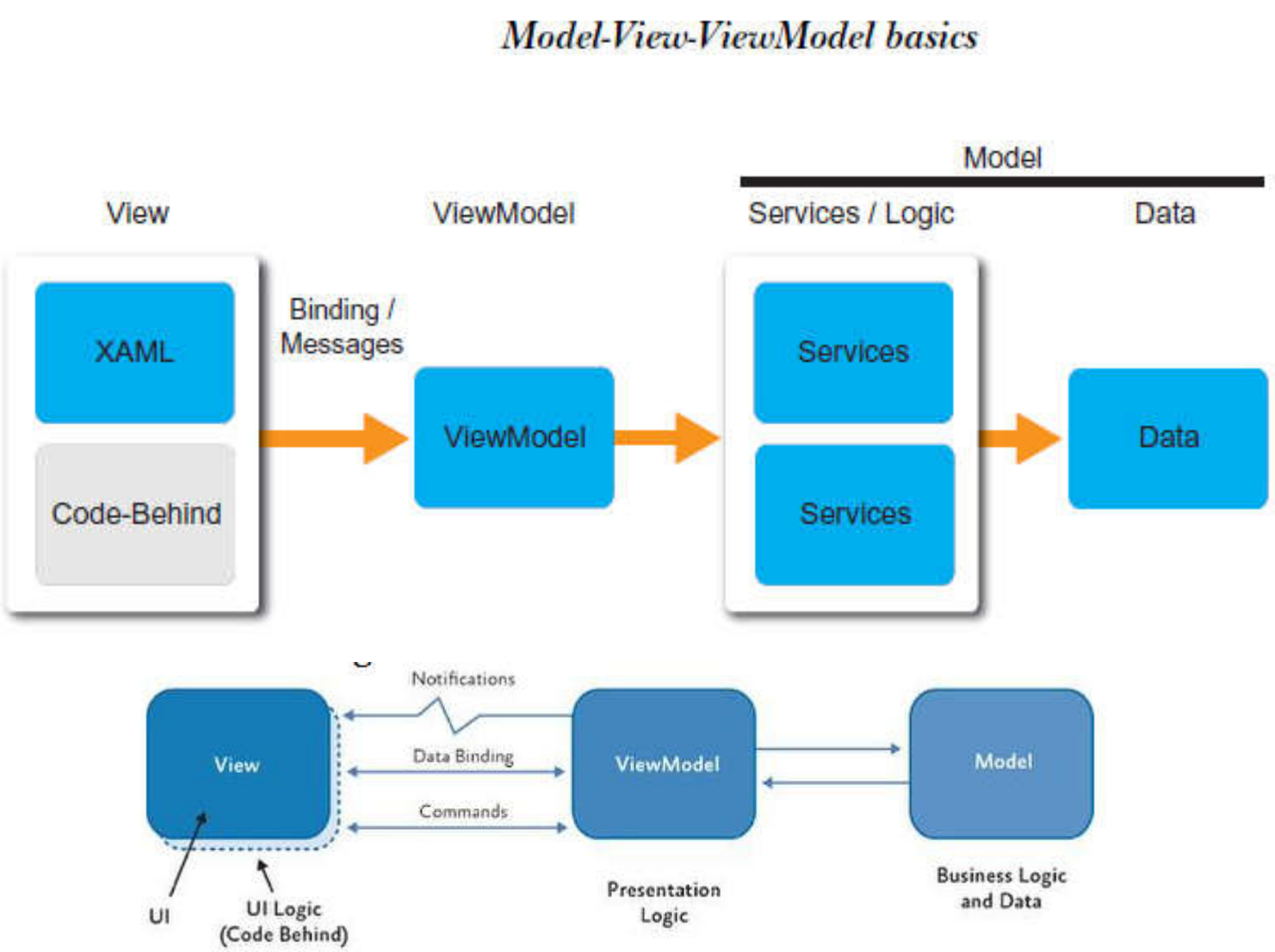


MVC

1. Separa el acceso a la lógica desde la presentación.
2. Utiliza un controlador para monitorear el control de la aplicación
3. Presenta datos en diferentes vistas y soporta actualizaciones a través de múltiples interacciones
4. Si se necesita adicionar otro tipo de cliente a la aplicación, puede ser agregado
5. Favorece que los componentes sean reutilizados

1. Debido a la separación de la lógica, usualmente se incrementa el número de clases de las que se debe dar seguimiento
2. Normalmente en etapas tempranas del desarrollo implica mas tiempo de desarrollo

MVVM


Model-View-ViewModel basics
Model
View
ViewModel
Services / Logic
Data
XAML
Code-Behind
Binding / Messages
ViewModel
Services
Services
Data
Notifications
Data Binding
Commands
View
ViewModel
Model
UI
UI Logic (Code Behind)
Presentation Logic
Business Logic and Data

MVVM

Parte	Descripción
Model	El modelo del negocio o modelo de la aplicación. Contiene los servicios (web services, servicios negocio, servicios lógicos, etc) acceso a datos, etc
View	El código de los elementos gráficos y el code-behind con la sola responsabilidad de interacción con el usuario. Solo debería haber código de la Vista propiamente. Típicamente con suficiente conocimiento de la estructura del ViewModel para hacer bind pero no sabe nada más del sistema
ViewModel	Fachada o interface entre la Vista y el Modelo. Parte entidad, parte fachada y parte control. Contiene lógica propia mínima. Utiliza el Bind para hacer push/pull de los datos y comandos para llamar a los métodos. No tiene conocimiento de la estructura de la vista