SOFTWARE DESIGN AND ARCHITECTURE

INSTRUCTOR: M FAIZAN KHAN

Review

Implicit Asynchronous Communication Software Architecture

Outline

Interaction-Oriented Software Architectures

- MVC

Interaction-Oriented Software Architectures

The interaction-oriented software architecture decomposes the system into three major partitions:

- data module,
- control module, and
- view presentation module.

Each module has its own responsibilities.

The key point of this architecture is its separation of user interactions from data abstraction and business data processing.

Interaction-Oriented Software Architectures

The data module provides the data abstraction and all core business logic on data processing.

The view presentation module is responsible for visual or audio data output presentation and may also provide user input interface when necessary.

The control module determines the flow of control involving view selections, communications between modules, job dispatching, and certain data initialization and system configuration actions.

Model-View-Controller (MVC)

What is MVC?

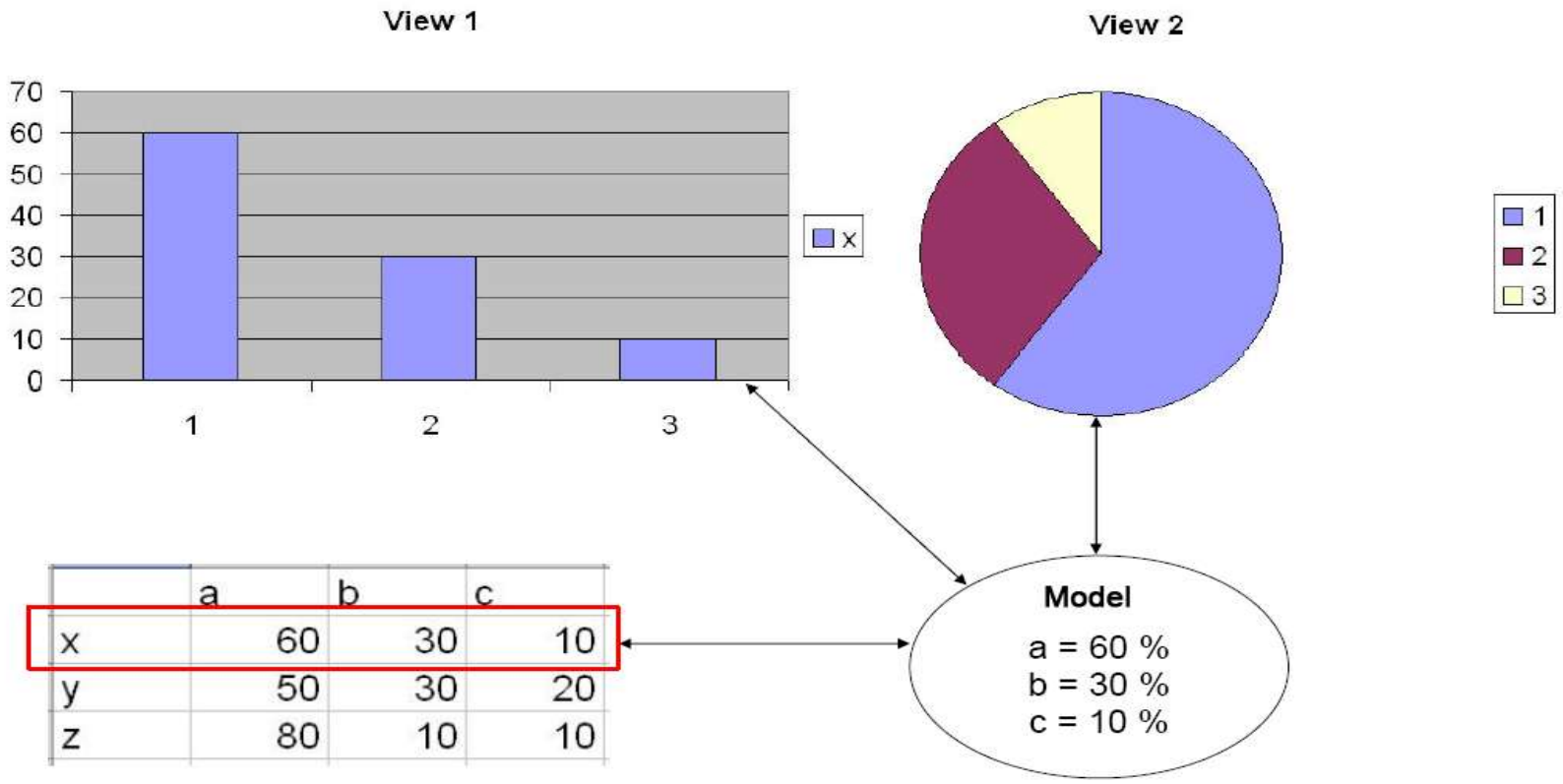
Architectural design pattern which works to separate data and UI for a more cohesive and modularized system

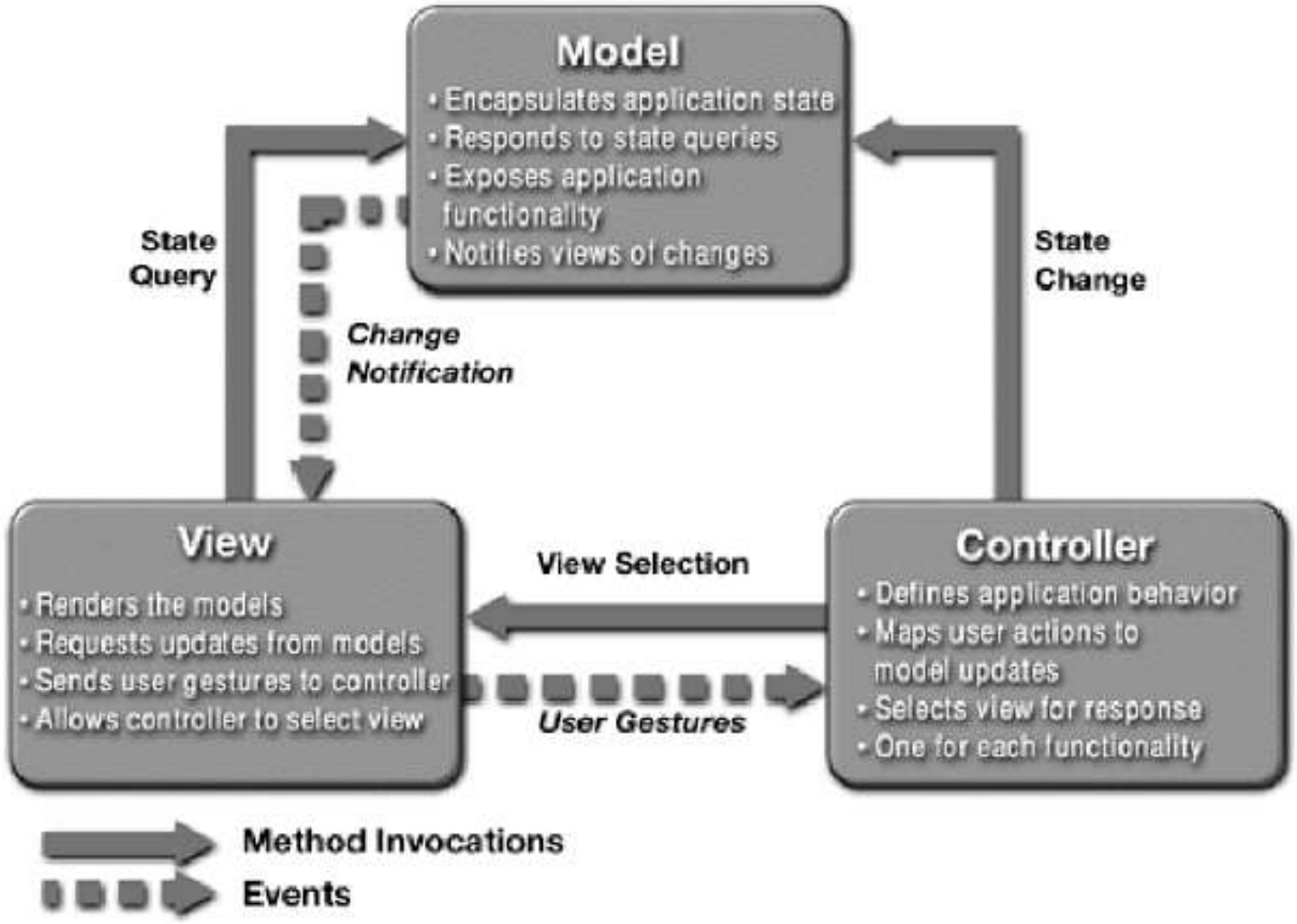
Model-View-Controller (MVC)

The MVC pattern was originally proposed in the 1980s as an approach to GUI design that allowed for

- multiple presentations of an object and
- separate styles of interaction with each of these presentations.

MVC – general example




Model
• Encapsulates application state
• Responds to state queries
• Exposes application functionality
• Notifies views of changes
State Query
State Change
Change Notification
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

Model

The domain-specific representation of the information on which the application operates.

The model is another name for the application logic layer

Application (or domain) logic adds meaning to raw data

Many applications use a persistent storage mechanism (such as a database) to store data.

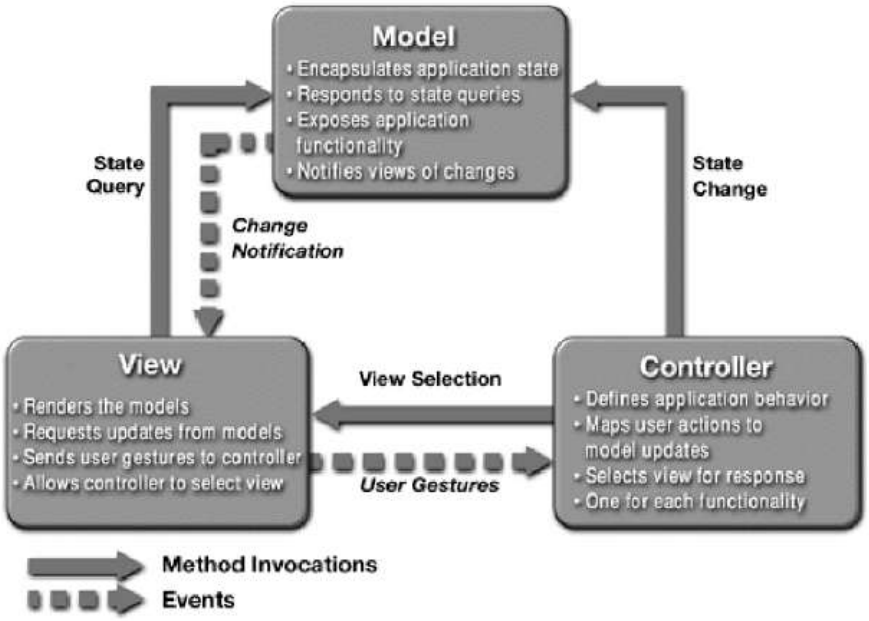
View

Renders the model into a form suitable for interaction, typically a user interface element.

MVC is often seen in web applications, where the view is the HTML page and the code which gathers dynamic data for the page.

Controller

Processes and responds to events, typically user actions, and may invoke changes on the model and view.



Example Control Flow in MVC

Though MVC comes in different flavours, the control flow generally works as follows:

1. The user interacts with the user interface in some way (e.g., user presses a button)

2. A controller handles the input event from the user interface, often via a registered handler or callback.

3. The controller accesses the model, possibly updating it in a way appropriate to the user's action (e.g., controller updates user's shopping cart).

Example Control Flow in MVC

4. A view uses the model to generate an appropriate user interface (e.g., view produces a screen listing the shopping cart contents).

The view gets its own data from the model.

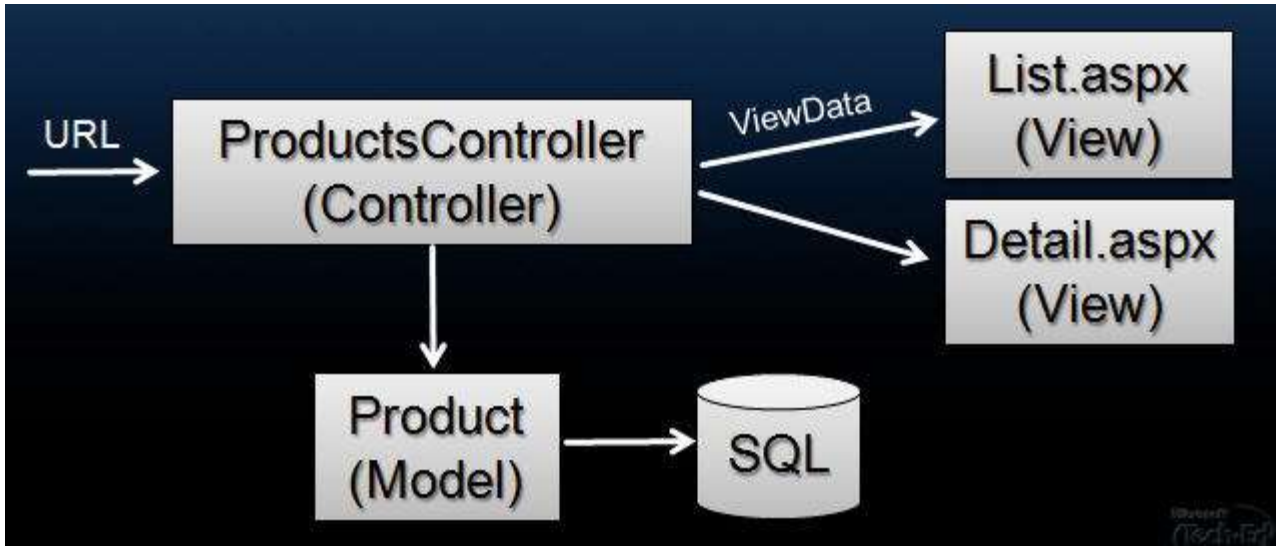
The model has no direct knowledge of the view.

5. The user interface waits for further user interactions, which begins the cycle anew.

Example:

Google maps

Model 2 and ASP.NET



Benefits:

Multiple views synchronized with same data model.

Easy to change or plug in new interface views, allowing updating of interface views with new technologies without overhauling the rest of the system.

Very effective for developments if graphics, programming, and database development professionals are working in a team in a designed project.

Limitations:

Not suitable for agent-oriented applications such as interactive mobile and robotics applications.

Multiple pairs of controllers and views based on the same data model make any data model change expensive.

The division between the View and the Controller is not clear in some cases.

Summary

Interaction-Oriented Software Architectures

- MVC