

Lecture 7:

Design a Web Page Using

MVC and Other Design

Approaches

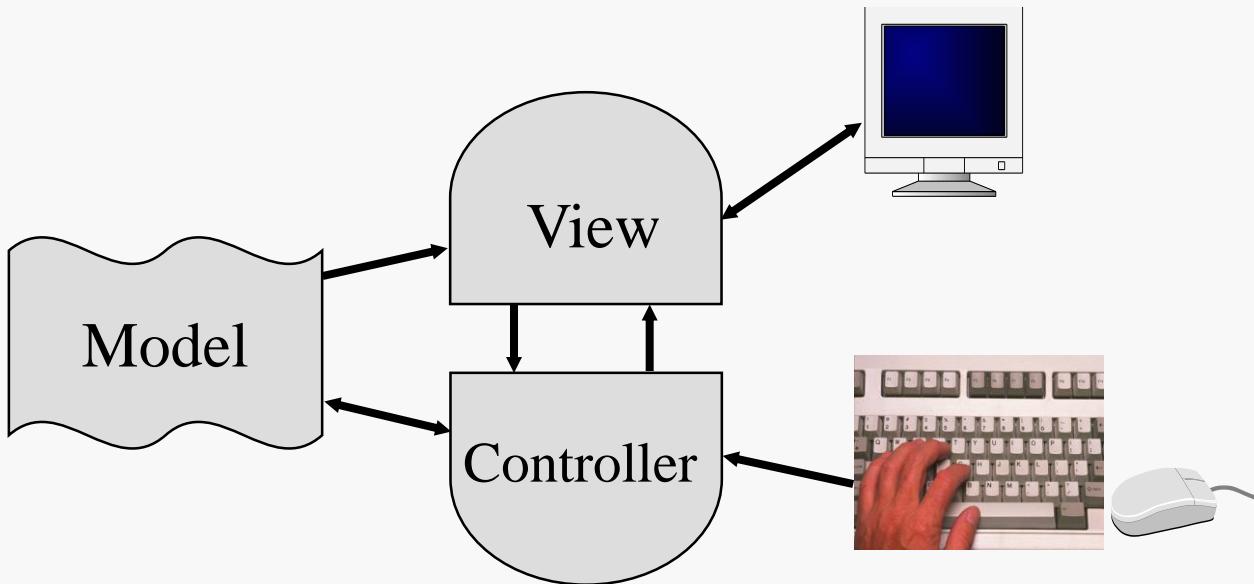
Relation to CLOs

The following Lecture is related to the CLO_2, which is as:

CLO_2: Evaluate the several web technologies and application architectures.

What is MVC ?

- **Model-View-Controller** (MVC) is a software architectural pattern which separates the representation of information from the user's interaction with it.
- The pattern isolates business logic from input and presentation, permitting independent development, testing and maintenance of each.



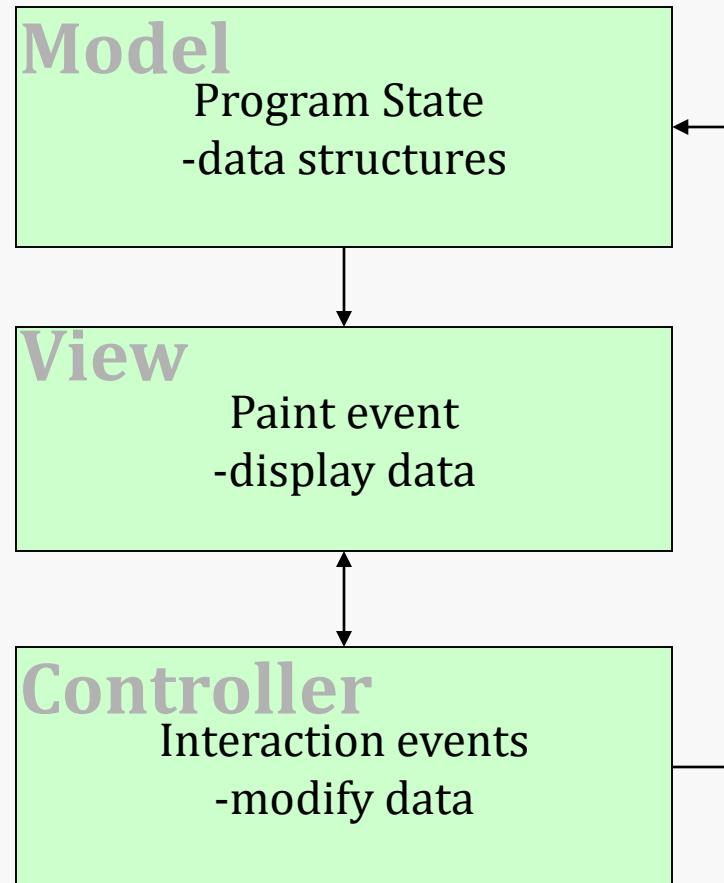
What is MVC?

- ❖ Model represents the data model
 - “*Manages behavior and data of the application domain*”
- ❖ View represents the screen(s) shown to the user
 - “*Manages the graphical and/or textual output to the portion of the bitmapped display that is allocated to its application*”
- ❖ Controller represents interactions from the user that changes the data and the view
 - “*Interprets the mouse and keyboard inputs from the user, commanding the model and/or the view to change as appropriate*” (Burbeck)

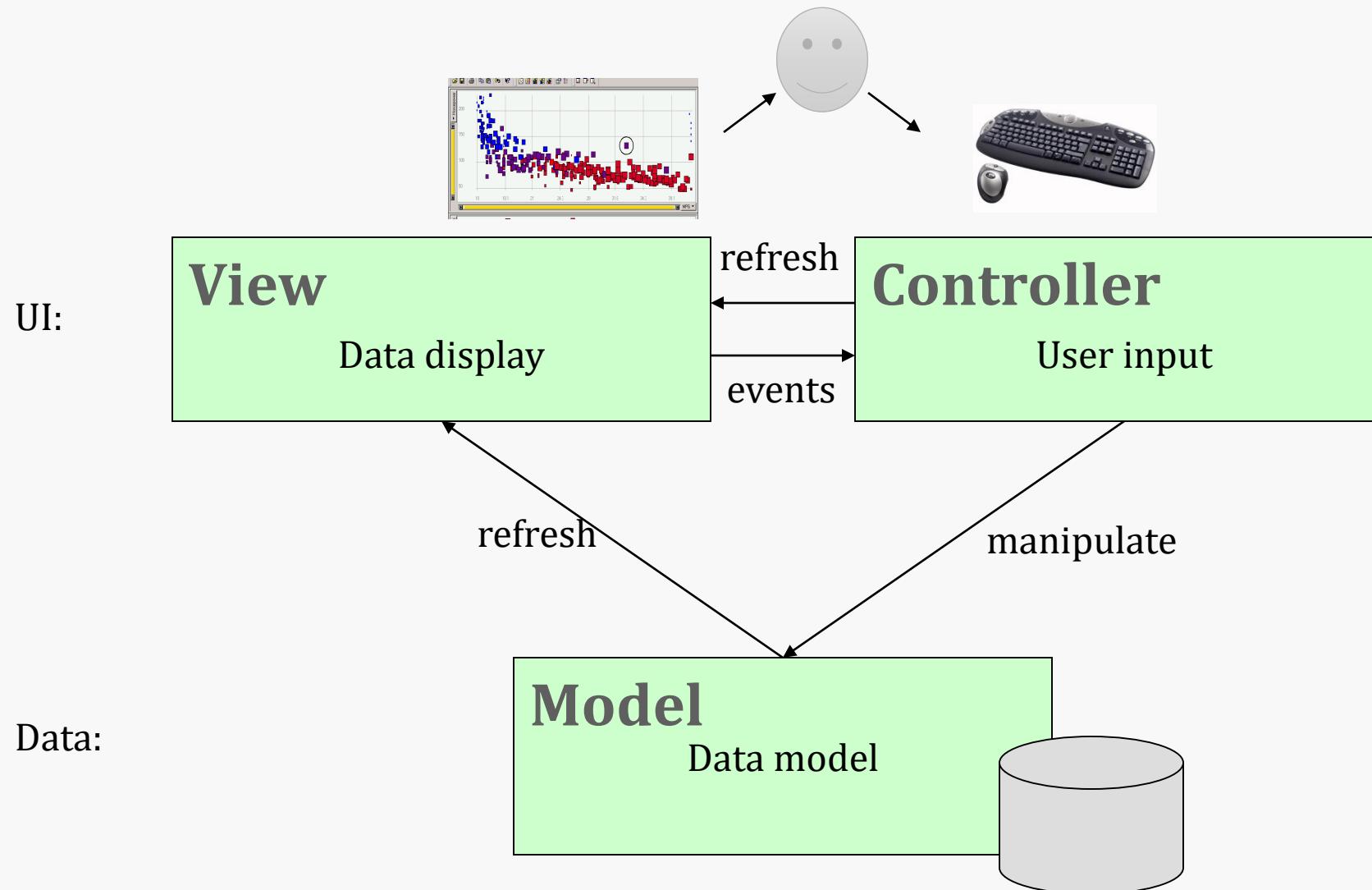
What is MVC?

- An MVC application is a collection of model/view/controller triplets .
- The **model** consists of **application data, business rules, logic and functions**
- A **view** can be any **output representation** of data such as a chart or diagram.
- Each **model** is associated with **one or more views** suitable for presentation.
- **Multiple views** for the same data are possible.
- When a **model changes** its state, it notifies its **associated views** so they can refresh.
- The **controller** is responsible for initiating change requests and providing any necessary data inputs to the **model**.

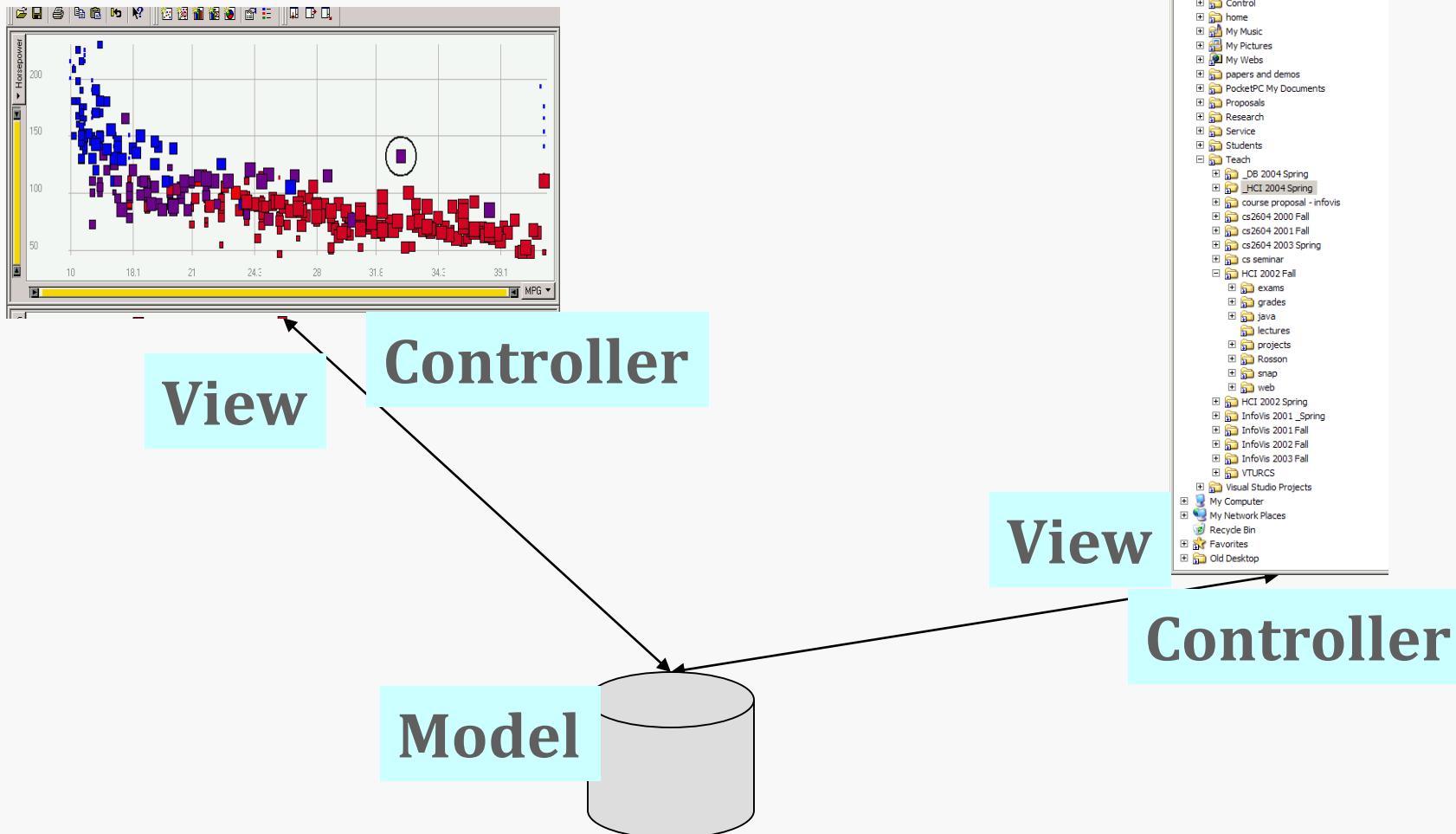
Model-View-Controller (MVC)



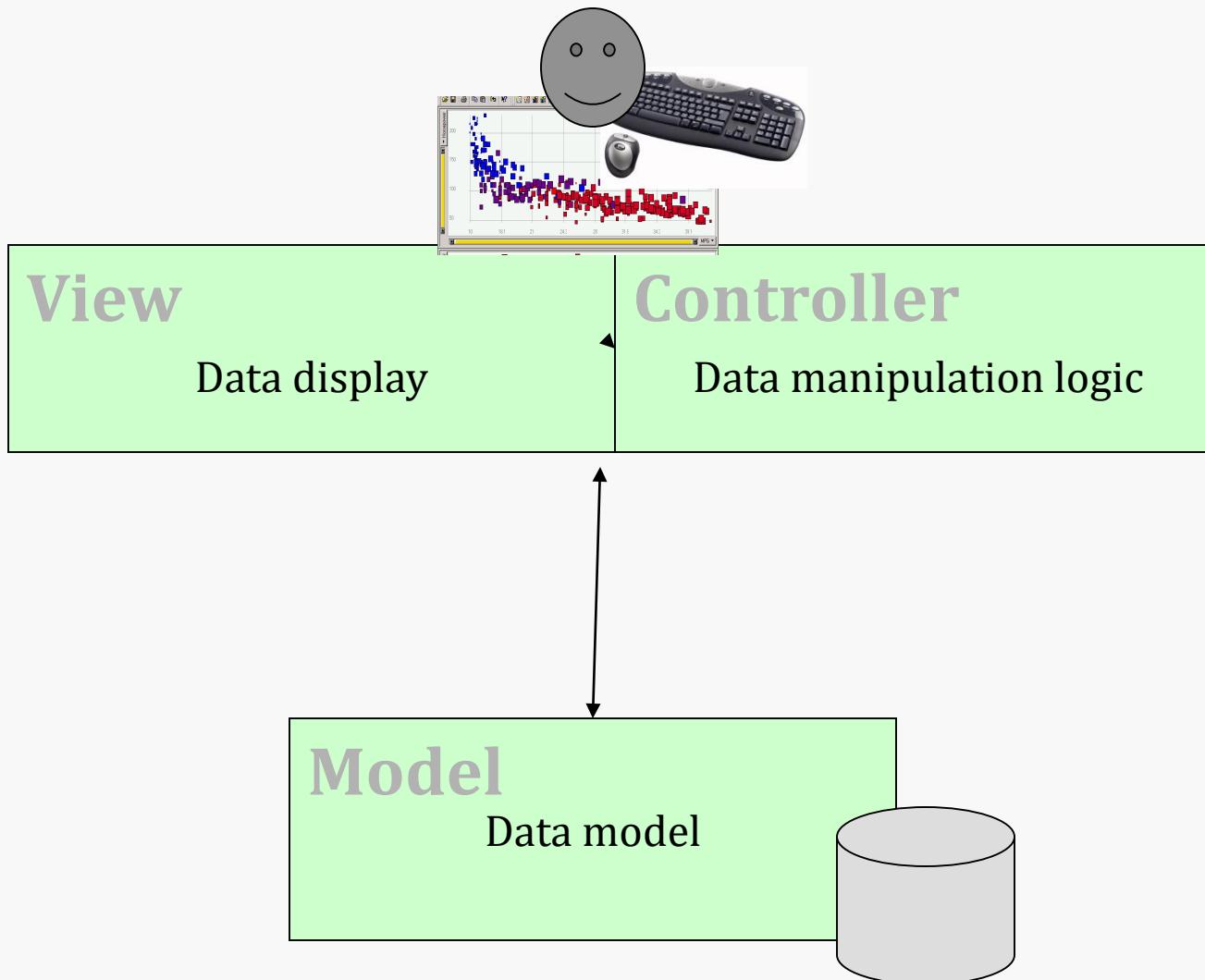
Model-View-Controller (MVC) diagram



Multiple Views



Typical real-world approach



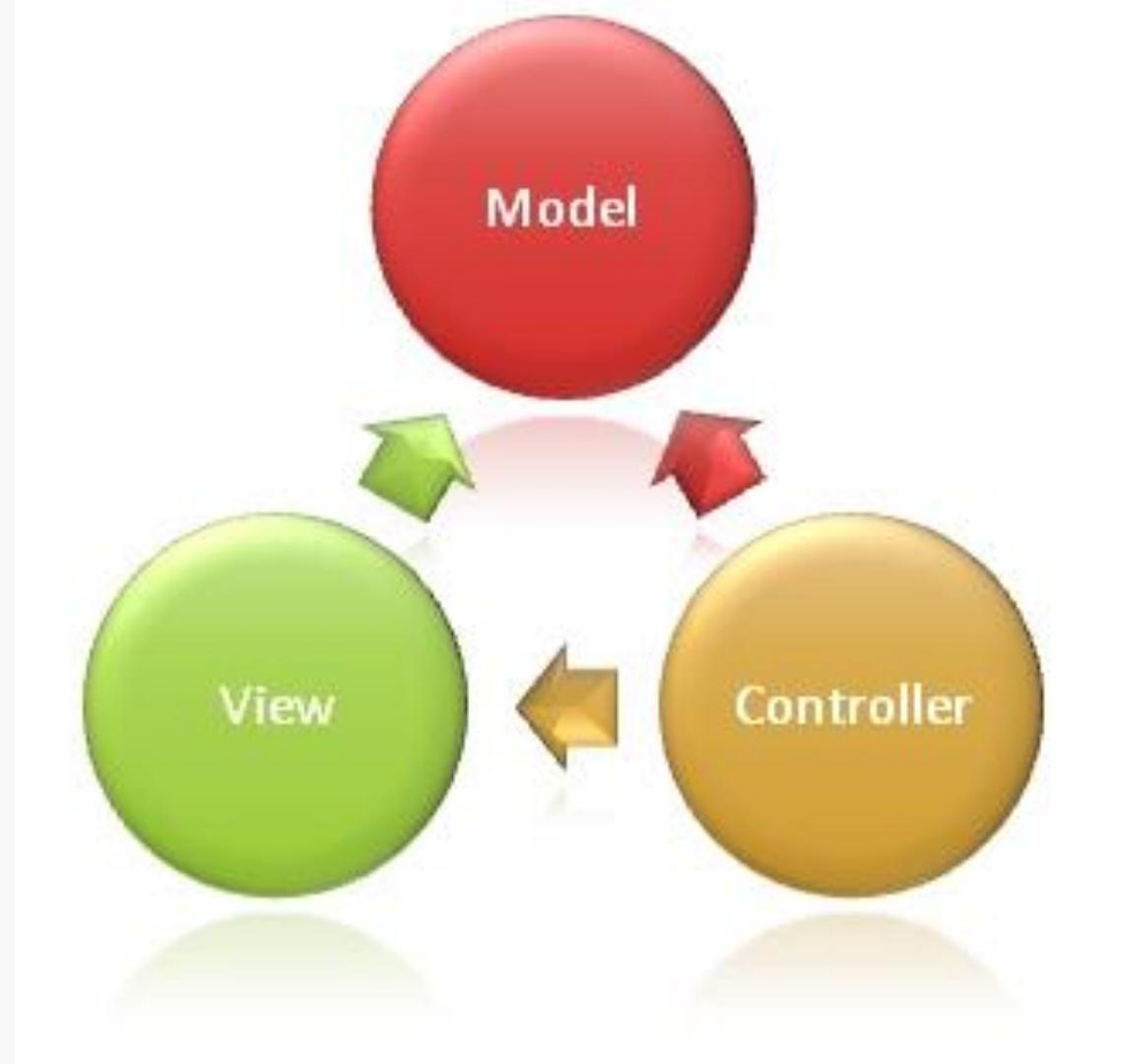
How did it come about?

- Presented by Trygve Reenskaug in 1979
- First used in the Smalltalk-80 framework
 - *Used in making Apple interfaces (Lisa and Macintosh)*

MVC Programming Model

The MVC model defines web applications with 3 logic layers:

- The business layer (Model logic)
- The display layer (View logic)
- The input control (Controller logic)



MVC Programming Model

- **The Model** is the part of the application that handles the **logic** for the application data. Often model objects retrieve data (and store data) from a database.
- **The View** is the parts of the application that handles the **display** of the **data**. Most often the views are created from the model data.
- **The Controller** is the part of the application that handles **user interaction**. Typically, controllers read data from a view, control user input, and send input data to the model.

MVC Components Interaction

MVC is a framework for building web applications using a MVC (Model View Controller) design:

Controller:

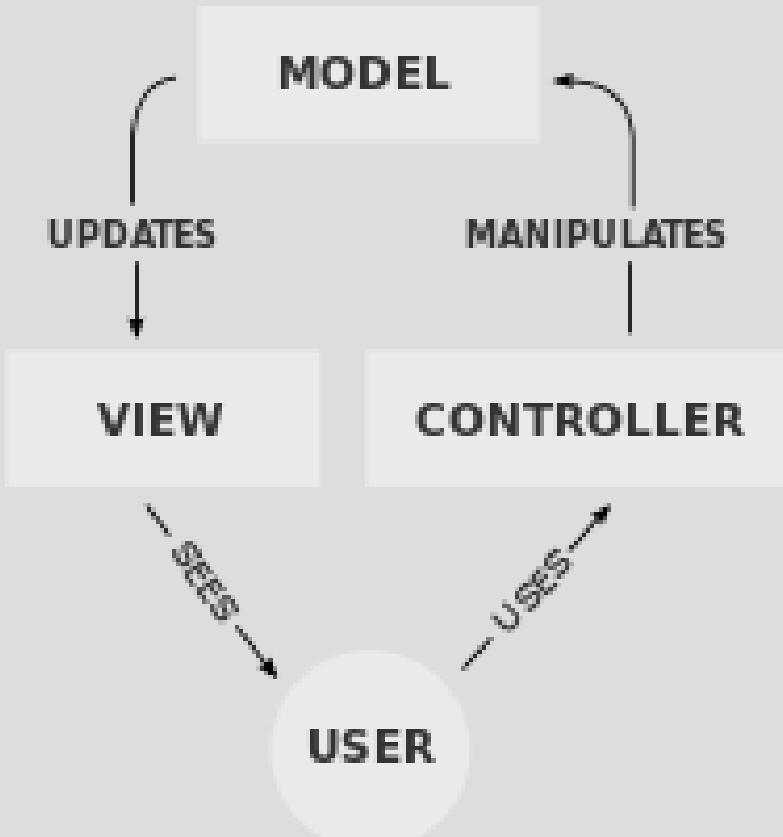
- A controller sends commands to the model to update the model's state.
- It can also send commands to its associated view to change the view's presentation of the model

Model

- A model notifies its associated views and controllers when there has been change in its state.
- This notification allows the views to produce the updated output.

View

- A view requests information from the model that it needs for generating an output representation to the user



MVC COMPONENTS INTERACTION

MVC Programming Model

- The MVC separation helps you manage complex applications, because you can focus on one aspect at a time.
- For example, you can focus on the view without depending on the business logic. It also makes it easier to test an application.
- The MVC separation also simplifies group development. Different developers can work on the view, the controller logic, and the business logic in parallel.

Pattern Description

- Model – view – controller is both an architectural pattern and a design pattern depending on where it is used.
- MVC can be used as :-
 1. *Architectural Pattern – high level strategy that concerns large scale components*
 2. *Design Pattern - influenced by programming language*

As Architectural Pattern

- MVC is seen in web applications where view is the HTML page.
- The controller is the code which gathers dynamic data and generates content within the HTML.
- The model is represented by the actual content is stored in database and the business rules that transfer the content based on user actions.

As Architectural Pattern

The Control flow is as follows :-

- The user interacts with the user interface in some way (e.g., by pressing a mouse button).
- The controller handles the input event from the user interface.
- The controller notifies the model of the user action, possibly resulting in a change in the model's state.

As Architectural Pattern

The view and model flow is as follows :-

- A view uses the model indirectly to generate an appropriate user interface (for example, the view lists the shopping cart's contents).
- The view gets its own data from the model. The model and controller have no direct knowledge of the view.
- The user interface waits for further user interactions, which restarts the cycle.

As a Design Pattern

- When considered as a design pattern, MVC is semantically similar to the Observer Pattern.
- Model:- Is the domain-specific representation of the data on which the application operates. Domain logic adds meaning to raw data (for example, calculating whether today is the user's birthday, or the totals, taxes, and shipping charges for shopping cart items).

As a Design Pattern

- View:- Transforms the model into a form suitable for interaction, typically a user interface element. Multiple views can exist for a single model for different purposes.
- Controller:- Processes and responds to events (typically user actions) and invoke changes on the model.

How MVC works

- The view receives user input and passes it to the controller.
- The controller receives user input from the view.
- The controller modifies the model in response to user input (or, in some cases, the controller modifies the view directly and does not update the model at all).

How MVC works

- The model changes based on an update from the controller.
- The model notifies the view of the change.
- The view updates the user interface, (i.e., presents the data in some way, perhaps by redrawing a visual component or by playing a sound).

Class Responsibilities in MVC

■ Responsibilities of the Model

- *Store data in properties*
- *Notify views of state changes*
- *Implement application logic*

Class Responsibilities in MVC

■ Responsibilities of the View

- *Create interface*
- *Update interface when model changes*
- *Forward input to the controller*

■ Responsibilities of controller

- *Translate user input into changes in the model.*

Advantages of MVC

- **Re-use of Model components.** The separation of model and view allows multiple views to use the same model.
- An application's model components are easier to implement, test, and maintain, since all access to the model goes through these components.
- **Easier support for new functionality:-** To add a new functionality, you simply write a view and some controller logic and add it to the existing application.
- Because the model is self-contained and separate from the controller and the view, it's much less painful to change your data layer or business rules.
- If you switch databases, say from MySQL to Oracle, you need only alter your model without affecting the other models in the view and the controller.

Disadvantages of MVC

- MVC requires significant planning.
- It introduces a deeper level of complexity that requires attention to detail.
- You'll have to spend a good amount of time thinking about the way the parts of your application will interact.
- Also, the separation between the model and view can sometimes make debugging more difficult.
- Using MVC also means having more files to manage than you would otherwise.
- MVC is not suitable for small applications and even many medium-size ones.

System Example (MVC)

- Air line reservation system developed for the software engineering course.
- This systems runs on the model view controller design pattern.