GOF & Design Patterns in Smalltalk MVC

Designed by Rita Ganatra

What is Gang of Four (GOF)?

In 1994, four authors Erich Gamma, Richard Helm, Ralph Johnson and John Vlissides published a book titled Design Patterns - Elements of Reusable Object-Oriented Software which initiated the concept of Design Pattern in Software development.

These authors are collectively known as Gang of Four (GOF). According to these authors design patterns are primarily based on the following principles of object oriented design.

Design Patterns in Smalltalk MVC

The Model-View-Controller (MVC) is an **architectural pattern** which separates an application into three main groups of components: Models, Views, and Controllers. MVC is abbreviated as **Model View Controller is a design pattern created for developing** applications specifically **web applications.**

The **Model View Controller** (MVC) design pattern specifies that an application consist of a **data model, presentation information, and control information**. The pattern requires that each of these be separated into different objects. MVC is more of an architectural pattern, but not for complete application. MVC mostly relates to the UI / interaction layer(Command Line Interface,Menu-driven Interface.,Graphical User Interface,Touch Screen Graphical User Interface) of an application.

Design components

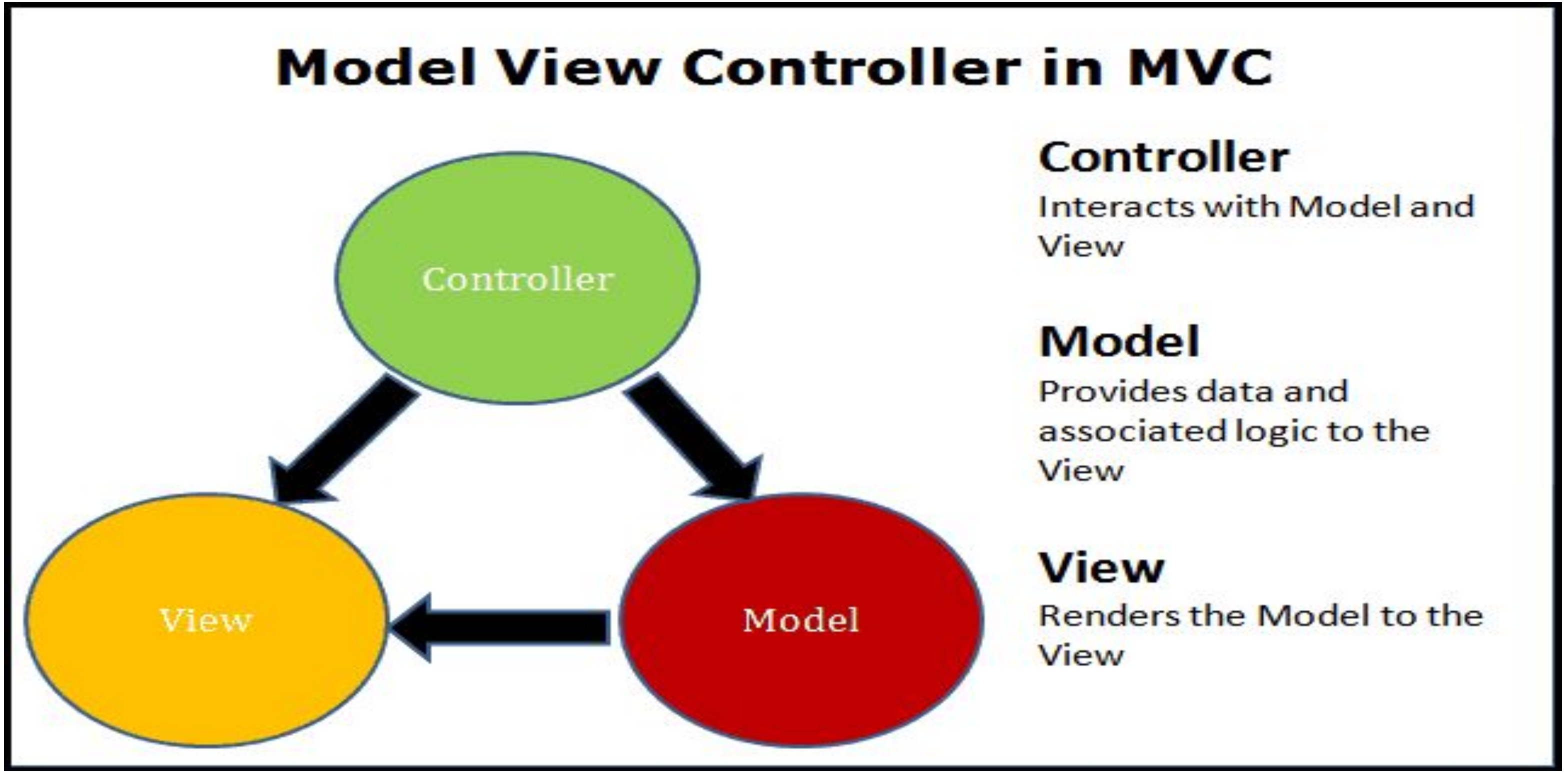
Three main groups of components: Models, Views, and Controllers.

- The **Model** contains only the pure **application data**, it contains no logic describing how to present the data to a user. (It’s just a data that is shipped across the application like for example from back-end server view and from front-end view to the database. In java programming, Model can be represented by the use of POJO (Plain-old-java-object) which is a simple java class.

- The **View presents the model's data** to the user. The view knows how to access the model's data, but it does not know what this data means or what the user can do to manipulate it. **View just represent, displays the application's data on screen.** View page are generally in the format of .html or .jsp in java programming (which is flexible).

- The **Controller** exists between the view and the model. It is where the **actual business logic is written.** It listens to events triggered by the view (or another external source) and executes the appropriate reaction to these events. In most cases, the reaction is to call a method on the model. Since the view and the model are connected through a notification mechanism, the result of this action is then automatically reflected in the view

Thus, MVC divides the application into three interconnected components, each with a distinct responsibility—Model for data handling, View for user interface, and Controller for user input processing.



Model View Controller in MVC

Controller

Interacts with Model and View

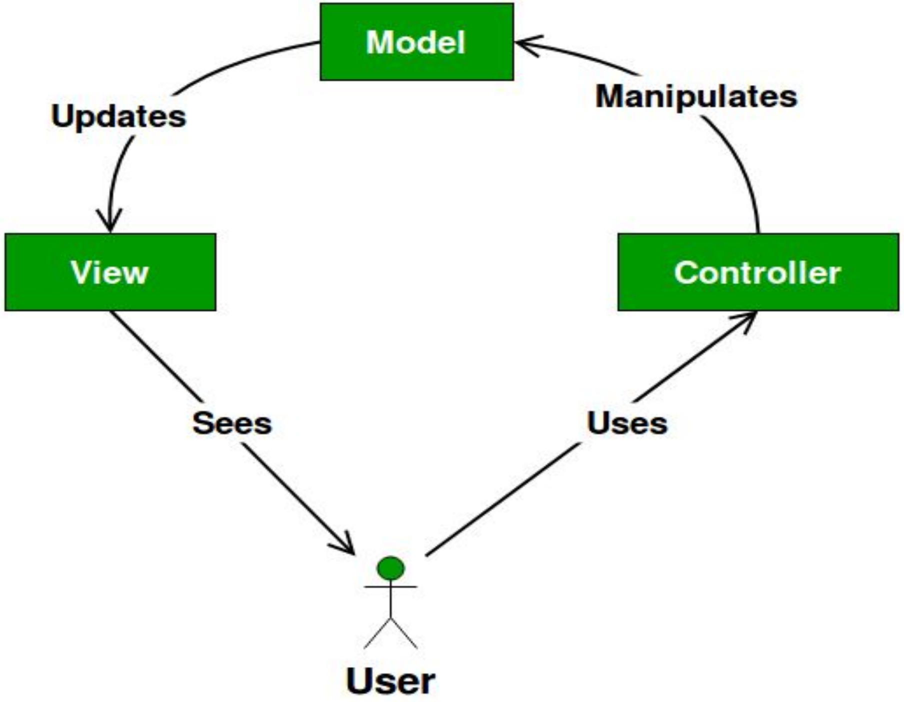
Model

Provides data and associated logic to the View

View

Renders the Model to the View

UML Diagram MVC Design Pattern



Advantages of Design Patterns in Smalltalk MVC

- **Multiple developers** can work simultaneously on the model, controller and views.
- MVC enables logical grouping of related actions on a controller together. The views for a specific model are also grouped together.
- Models can have multiple views.
- The overall components of an application are easily manageable & are less dependent on each other for proper functioning of application.

Disadvantages of Design Patterns in Smalltalk MVC

- The framework navigation can be complex because it introduces new layers of abstraction and requires users to adapt to the decomposition criteria of MVC.
- Knowledge on multiple technologies becomes the norm. Developers using MVC need to be skilled in multiple technologies.

Disadvantages of Design Patterns in Smalltalk MVC

- Design patterns can add a layer of complexity to the application. For small projects or simple applications, this might be unnecessary and could make the codebase harder to understand.
- For developers who are not familiar with design patterns or the MVC architecture, there can be a steep learning curve. This can slow down initial development and onboarding of new team members.(learning something new is particularly challenging and requires a significant amount of effort, time, and concentration to master.)
- Knowledge on multiple technologies becomes the norm. Developers using MVC need to be skilled in multiple technologies.

Thank You