MVC v/s MVP - How Common and How Different

Neeraj.Soni

Rate me: ★★★★☆ 4.29/5 (13 votes)

24 Jan 2013 CPOL

A comparative introduction to MCV and MVP

Introduction

Over the years, design patterns are being best practiced in IT and SE. Design Patterns are the solution guidelines to common problems identified in software applications. DPs are taken care of during application design or during refactor process. Most of the times, DPs are not followed because of lack of experience or knowledge. In case of MCV/MVP, lack of knowledge of difference between them is one of the fear factors for not choosing the right pattern. This article would try to differentiate between the usage of MVC and MVP.

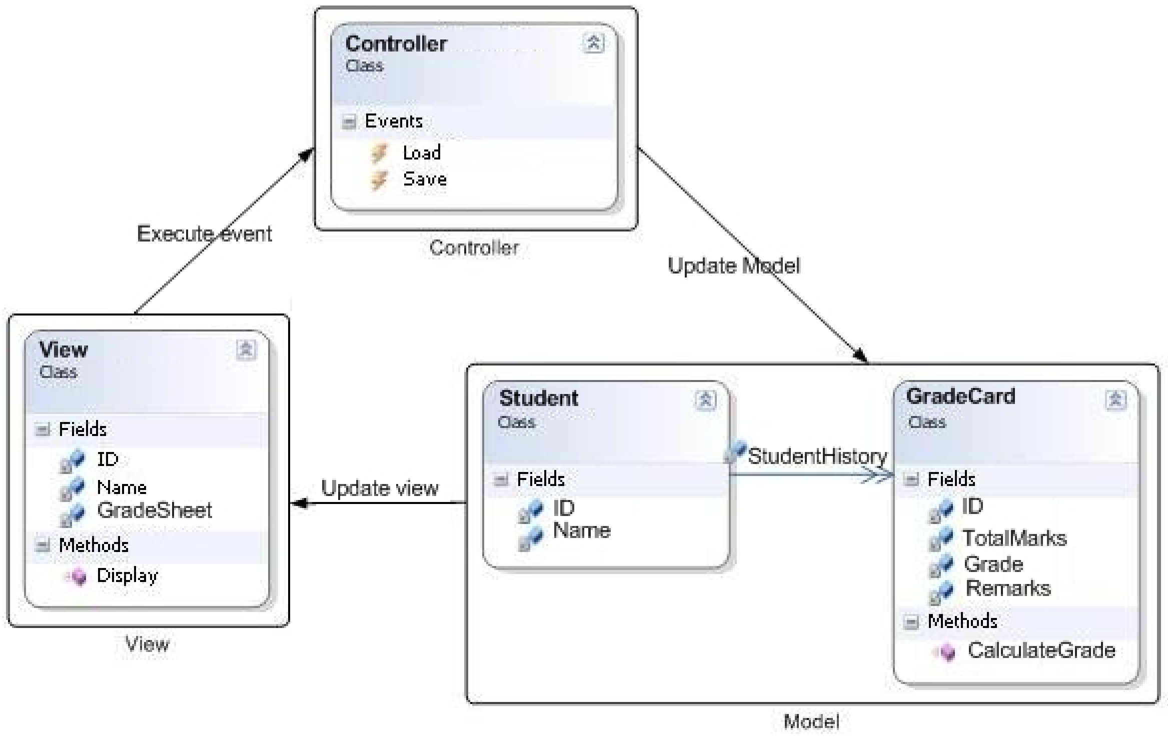
Before going into how MVC/MVP work and what are the key benefits of either of the two, a note that can be made is that both of them are being practiced for many years to achieve the OO principle of separation of concern for UI Layer and Business Layers. Both of them are UI presentation patterns.

A number of frameworks are provided to implement these patterns. For example:

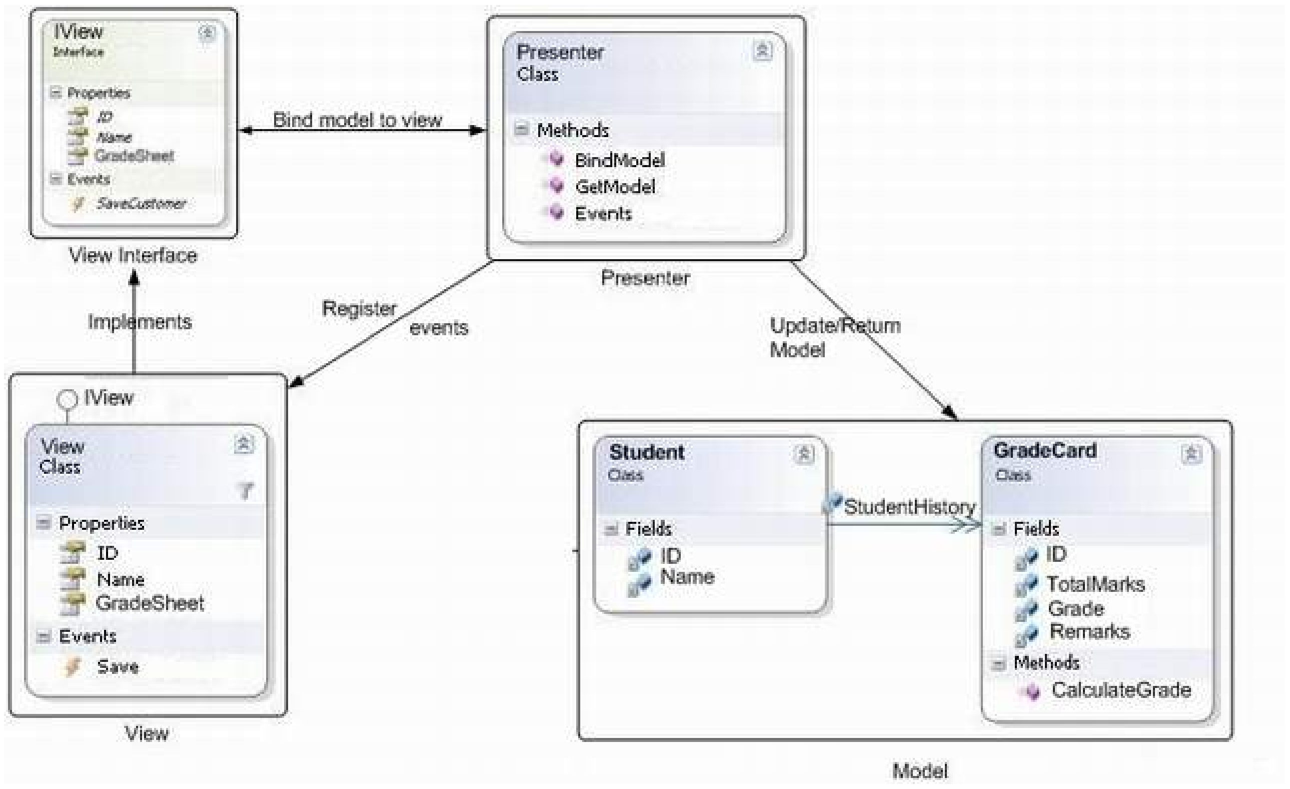
1. JAVA Struts
2. ROR (Ruby on Rails)
3. Microsoft Smart Client Software Factory (CAB)
4. Microsoft Web Client Software Factory
5. ASP.NET MVC Framework

DPs are the blueprints but not the solution itself. They should be used as a guideline for the implementation within the problem domain/space.

MVC - Class Diagram



MVP - Class Diagram



Differences

S.No.	MVC	MVP

S.No.	MVC	MVP
1	UI Presentation Pattern focus on separation of view with Model	Based on MVC (UI Presentation Pattern)
2	Separation of responsibility between three components: 1. View - responsible for rendering UI elements 2. Controller - responsible for responding to view actions 3. Model - responsible for business behavior and state management	Separation of responsibility between four components: 1. View - responsible for rendering UI elements 2. View Interface - responsible for loose coupling between view and model 3. Presenter - responsible for view and model interaction 4. Model - responsible for business behavior and state management
3	The three components would interact with each other. Controller would sometime also be responsible to update view (like Front Controller Pattern)	Presenter can also interact with Controller to access Model
4	Fairly loose coupling	Comparatively high degree of loose coupling
5	Limited unit testing	Comparatively high degree of ease of unit testing
6	Controller is behavior based and multiple views can share single controller	Usually one view has one presenter (1-1 mapping). Multiple presenters would be associated with a complex view
7	Identifies which view to update	Presenter will update its associated view

Benefits

Any pattern has its pros and cons in association with the environment it is being implemented in. It is not always advisable to implement the patterns everywhere in the application. There are benefits of MVC/MVP usage:

1. Code reuse - Separation of concern principle will provide code reusability as the design will have a proper domain model and business logic in its logical unit.
2. Adaptable design - A good design is closed for changes and open for additions. Isolated code in presenter/controller, domain model, view, and data access provide the freedom of choosing a number of views and data sources.
3. Layering - MVC/MVP forces to separate data access login for the other layers and various other patterns would be opted to implement the data access layer.
4. Test driven approach - Isolated implementation allows to test each component separately. Especially in MVP pattern that uses interface for a view, it is a true test driven approach.

Drawbacks

1. Higher complexity
2. Extra learning curve
3. Experience and knowledge makes a difference in the right implementation
4. Not suitable for simple and small solutions

Credits

I thank to Todd Snyder who had written a very good article and explained it well for everyone. All the credit of this article also goes to him only.

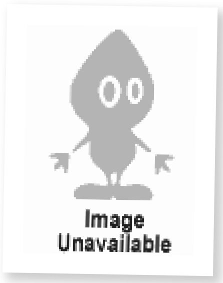
History

- 27th November, 2008: Initial post

License

Share

About the Author



Neeraj.Soni

Software Developer (Senior)

India

neerajsoni@gmail.com

Comments and Discussions

You must Sign In to use this message board.

Search Comments

First Prev Next

Strongly Disagree with few things
Ranjan.D 12-Jan-13 11:09

...
Jayson Ragasa 30-Nov-11 15:49

My vote of 5
radumi 5-Aug-10 18:48

My vote of 2
Greg Olmstead 12-Dec-08 8:23

Controller in MVP?
zeneclectic 1-Dec-08 7:16

Re: Controller in MVP? [modified]
Neeraj.Soni 1-Dec-08 22:04

My vote of 2
tf124 28-Nov-08 16:01

My vote of 1
Puchko Vasili 27-Nov-08 5:48

My vote of 1
Sadadevguru 27-Nov-08 3:25

Refresh 1

General News Suggestion Question Bug Answer Joke Praise Rant Admin

Use Ctrl+Left/Right to switch messages, Ctrl+Up/Down to switch threads, Ctrl+Shift+Left/Right to switch pages.

Permalink
Advertise
Privacy
Cookies
Terms of Use

Layout: fixed | fluid

Web03 2.8.2022.02.25.1