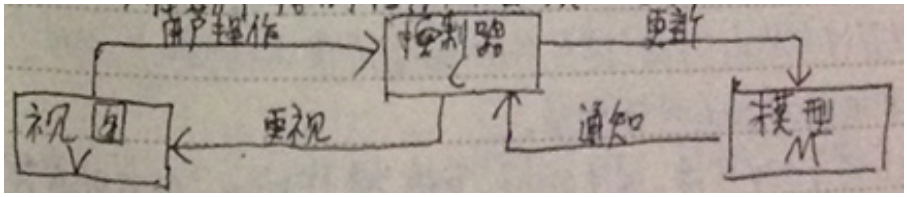
MVC、MVVM 浅谈

1. MVC

(1) 简介：模型－视图－控制器(Model-View-Controller,MVC)是 xerox PARC 在 20 世纪 80 年代为编程语言 smalltalk－80 发明的一种软件设计模式，至今已广泛用于用户交互应用程序中。在 ios 开发中 MVC 的机制被使用的淋漓尽致，充分理解 ios 的 MVC 模式，有助于我们程序的组织合理性。



(2) 模型对象

模型对象封装应用程序的数据，并定义操控和处理该数据的逻辑和运算。例如：模型对象可能表示游戏中的角色或地薄中的联系人。用户在视图层中所创建或修改数据的操作，由控制器对象传达出去，最终会创建或更新模型对象。模型对象更改时(如网络请求了新数据)，它通知控制器对象，控制器对象。

(3) 视图对象

视图对象是应用程序中用户可以看见的对象。视图对象知道如何将自己绘制出来并可能对用户操作作出事件响应(UIControl 及其子类)。视图对象的主要目的，就是显示来自应用程序模型对象的数据，并使该数据可被编辑。尽管如此，在 MVC 应用程序中，视图对象通常与模型对象分离。在 ios 应用程序开发中，所有的控件、窗口等都继承自 UIView，对应 MVC 中的 V。UIView 及其子类主要负责 UI 的实现，而 UIView 所产生的事件可才用委托的方式交给视图控制器去实现。

(4) 控制器对象

在应用程序的一个或多个视图对象和一个或多个模型对象之间，控制器对象充当媒介。控制器对象因此是同步管道程序，通过它，视图对象了解模型对象的更改反之亦然。控制器对象还可以为应用程序执行设置和协调任务，并管理其他对象的生命周期。控制器对象解释视图对象中进行的用户操作，并将新的或更改过的数据传给模型对象。当模型对象修改时，一个控制器对象会将新的模型数据传达给视图对象，以便视图对象可显示它对应不同的 UIView，有相应 UIViewController，对应 MVC 中的 C，例如 ios 常用的 UITableView，它所对应的 controller 就是 UITableViewController

(5)注意事项

1) Model 和 View 永远不能互相通信，只能通过 Controller 传递

2) Controller 可以直接与 Model 对话(读写 Model 属性调用 Model 接口)，Model 通过通知和 kvo 机制与 Controller 间接通信

3) Controller 可以直接与 View 对话，通过 outlet，直接操作 view，outlet 直接对应道 view 中的控件，view 通过 action 向 controller 报告事件的发生，controller 是 view 的直接数据源(数据可能是 controller 从 model 中取得并已经加工过了)。Controller 是 view 的代理(delegate)，以同步 view 与 controller

2. MVVM

(1)简介：在 ios 应用中日益增长的重量级视图控制器问题：在典型的 MVC 应用中，许多逻辑被放在 controller 里。有些确实属于 controller，但更多的是表示逻辑，为了不让控制器日益复杂，便于测试管理，出现了 MVVM。它其实是一个 MVC 的增效版，并将表示逻辑从 controller 移除放入新对象里，即 viewModel 里

(2)Model 层：存放数据，并定义数据操控和处理该数据的逻辑和运算

(3)ViewModel 层：view 和 model 层的粘合剂，它是一个放置用户输入验证逻辑、视图显示逻辑、发起网络请求和其他各式各样代码的好地方。简而言之，即把 原来 viewController 层的业务逻辑和页面逻辑等剥离出来放到 viewModel 层

(4)View 层：即视图和视图控制器层，任务是从 viewModel 层取数据，然后显示。