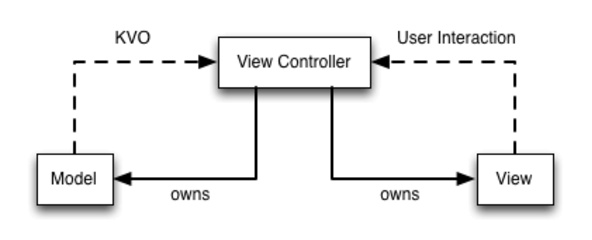
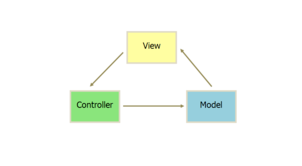
MVVM自我了解

我们现在经常用到的时MVC模式 Model-View-Controller ，Model 呈现数据，View 呈现用户界面，而 View Controller 调节它两者之间的交互。



**MVC架构：**  
View：对应于布局文件  
Model：业务逻辑和实体模型  
Controllor：对应于Activity

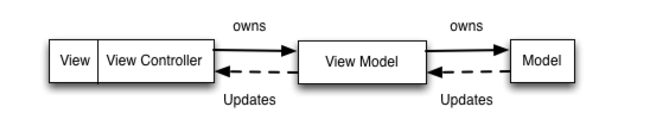


View可以与Model直接交互。

Controller是基于行为的，并且可以被多个View共享。

可以负责决定显示哪个View。

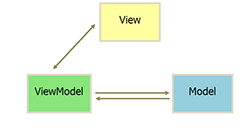
MVVM模式



MVVM是Model-View-ViewModel的简写

**MVVM架构：**  
Model：代表你的基本业务逻辑  
View：显示内容  
ViewModel：将前面两者联系在一起的对象

view model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种各样的代码的极好的地方。



一个ViewModel接口提供了两个东西：动作和数据。动作改变Model的下层（click listener，监听文字改变的listener等等），而数据则是Model的内容。

去年的I/O大会上谷歌介绍了一个非常好用的新框架DataBinding，该框架可以让你将view和一个对象的对field绑定。当field更新的时候，framework将收到通知，同时view也会自动更新。其语法和使用方式和 JSP 中的 EL 表达式非常类似。

在MVVM中，ViewModel在改变内容之后通知binding framework内容发生了改变。然后framework自动更新和那些内容绑定的view。这两个组件只是通过ViewModel松耦合在一起。这种设计模式之所以好用和方便，除了明显智能化了的View之外，还方便了测试。因为ViewModel不在依赖于View了，你可以在没有View的情况下也能测试ViewModel。在合适的依赖注入的帮助下，测试就会变得非常简单。

**MVVM的优点**

**1. 低耦合**。视图（View）可以独立于Model变化和修改，一个ViewModel可以绑定到不同的"View"上，当View变化的时候Model可以不变，当Model变化的时候View也可以不变。

**2. 可重用性**。你可以把一些视图逻辑放在一个ViewModel里面，让很多view重用这段视图逻辑。

<http://blog.ypli.xyz/ios/rac-mvvmshi-jian-er-wan-zheng-demoshi-jian>

<http://www.saitjr.com/ios/ios-framework-reactivecocoa-mvvmafnetworking-fmdb.html>

<http://www.cnblogs.com/brycezhang/p/3840567.html>

<https://github.com/leichunfeng/MVVMReactiveCocoa>