MVC模式在软件设计应用中的研究

周永平 宁夏人事考试中心，宁夏 银川 750004

摘要：随着软件开发技术的不断更新和改进，为了提高软件开发的效率并实现软件开发产业化，软件开发已步入软件产品的模块化和可复用性道路。而目前，设计模式就是实现软件模块化、提高软件可复用性和软件开发效率这一目标的一个行之有效的方法。所谓设计模式是人们在长期的软件开发实践中，总结出的一些解决特定问题的有效方法，并加以归纳和提炼，在遇到类似问题时，就可以直接复用以前的方法。因此，为了解决软件设计中的非功能性需求，促进软件开发的产品化，设计模式在软件设计中的研究及应用越来越受到人们的重视。

关键词：MVC模式；应用研究

中图分类号：TP31 **文献标识码：**A **文章编号：**1003-9767(2009)11-0058-02

The study of MVC pattern in software design applications

Zhou Yongping, Ningxia Personnel Testing, Ningxia, Yinchuan, 750004

Abstract: With continuously updated and improved of the software development technology, in order to improve the efficiency of software development and realize the industrialization of software development, software development has entered a modular software products and reusability of road.Currently, design patterns is to achieve software modularity, reusability, and software to improve software development efficiency of this goal an effective method. The so-called design patterns is that people in long-term software development practice, summed up some of the effective ways to solve specific problems, and to summarize and refined, in the face of similar problems, they can re-use the previous methods.Therefore, in order to solve software design of the non-functional requirements, to promote the product-oriented software development, design patterns in software design, research and application of more and more attention.

Keywords: MVC model; Applied Research

引言：

自上个世纪40年代软件诞生以来，虽然软件产业一直保持着高速发展的态势，但其状况仍不尽如人意。起初，人们把软件设计的重点放在数据结构和算法上。随着软件系统规模越来越大、越来越复杂，人们开始采用工程化的方法来开发软件，即使用软件工程的方法开发软件。但是随着软件危机程度的日益加剧，现有的软件工程方法已不能很好地满足人们的要求。

1. 设计模式

1.1 设计模式概述

设计模式最初来源于工程和建筑模式，软件设计模式的概念得益于建筑师Christopher Alexander的著作--《建筑的永恒之道》。在此书中作者认为：每一个设计模式描述了一个在我们周围不断重复发生的问题，以及该问题的解决方案。软件界吸收了这一成果，并不断深化。在1995年初，“四人组”(Gang of Four)出版了书《设计模式：可重用的面向对象软件的元素》，这本书包含了设计模式的一个基本目录，并且确定设计模式为软件学科中的一个新的领域。

当设计、构建不同的应用程序时，经常会遇到相同或者类似的问题，应用设计模式(Design Pattern) 就不需要每次遇到这些问题时重新寻找解决方案。设计模式的思想认为在系统设计这一层次上，软件开发可以抽象成一种模式，这种模式描述了我们在设计时不断重复遇到的问题，以及该问题的解决方案的核心内容。设计模式有利于帮助做出一些利于系统复用的选择，同时避免设计破坏系统复用性，在很大程度上改善软件结构，使开发具有更好的弹性和通用性。

1.2 MVC设计模式

有相互联系的简单设计模式组织在一起可形成的新模式，有可能提供一个整体的应用系统解决方案。MVC就是这样一种复合型设计模式。

MVC(Model-View-Controller/模型-视图-控制器)模式是在八十年代为编程语言Smalltalk-80发明的一种软件设计模式，早期它主要用于设计用户界面。然而MVC的思想却应用广泛，Microsoft的MFC基础类也遵循了MVC的思想。MVC模式目前已趋于完善，成为一种经典的面向对象的设计模式，它所使用的范围也不仅仅局限于用户界面的设计和实现，它体现的功能分离思想已经广泛应用到软件设计的其它层面上。

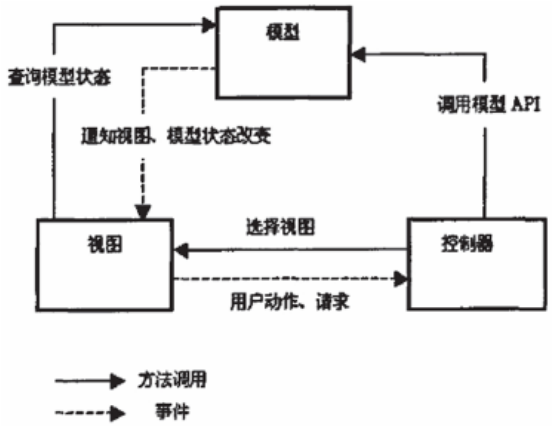
1.3 MVC模式的结构

MVC主要由三部分组成：Model(模型)，View(视图)，Controller(控制器)，各部分的职能如下：

(1)Model(模型)：模型部分，这是整个模式的核心。它封装了应用问题的核心数据、逻辑关系和业务规则，提供了完成问题的操作过程。

(2)View(视图)：用户视图部分，这是用户界面部分。与Web应用

图1 MVC模式各部分的协作方式



程序一样，主管应用程序与人之间的接口。一方面它为用户提供了输入手段，并触发应用逻辑运行；另一方面，它又将逻辑运行的结果以某种形式显示给用户。

(3)Controller(控制器)：控制器部分，该部分是用户界面与Model的接口。一方面它解释来自于View的输入，将其解释成为系统能够理解的对象，同时它也识别用户动作，并将其解释为对Model特定方法的调用；另一方面，它处理来自于Model的事件和Model逻辑执行的结果，调用适当的View为用户提供反馈。Controller的作用非常重要的，它起到了一个枢纽的作用，整个应用的走向，不论是页面之间的跳转还是选择具体的处理逻辑，就由它来控制。

1.4 MVC模式的工作原理

图1表现了MVC各部分的协同工作方式：

在MVC模式中，Controller接收使用者的消息，要求Model处理应用领域的资料；Model告诉View，让View知道Model的内容已更新；View接获通知并进行准备工作，就绪后才要求Model送来新内容并显示。

从上面的各部分的协同工作图中，我们可以看出：MVC的价值主要在于：1.从模型中分离视图2.从视图中分离控制器。基于这两个分离，MVC很好的解决了软件工程中如何使软件系统各模块之间最大限度地降低其复杂的耦合关系，以及系统显示逻辑和业务逻辑之间的矛盾(即用户界面的多变性和业务逻辑的相对不变性)，尽可能地提高系统的可维护性和扩展性。模型具有可移植性和伸缩性，模型的相对独立性使得它很容易被移植到新的平台工作，很容易被改变业务规则而不影响或者较小的影响视图和控制器。

用MVC分析应用程序的特性，有助于将一个应用程序分割为更易于重构的逻辑组件。MVC结构是用来分割对象间功能的一种有效方法，用于数据的维护和表示，从而降低系统模块间的耦合度。

2. MVC模式的优点和缺点

2.1 MVC模式的优点

MVC模式的显著优点是分离了数据和其表示，使得添加和删除一个用户的视图变得很容易，甚至可以在程序执行中动态进行用户的视图更新，同时使得模型和视图可以单独地进行开发，增加了程序的可维护性和可扩展性，并使得测试更为容易。

MVC模式的优点如下：

(1)设计清晰，易于维护：控制器和视图可以随着模型的扩展而进行相应的扩展，只要保持一种公共的接口，控制器和视图的旧版本也可以使用。当设计一个应用时，这种方式将使整个程序更加容易执行和维护。

(2)各个组件的独立性：基本上任何组件甚至整个模块都可以进行修改或者替换，模型、视图或控制器的程序改变不会都影响到其它方面。这样使得不同的组件和模块的开发能够同时进行。视图与控制器的通信方式允许更换视图和控制器，还可以根据需求动态的打开或关闭、甚至在运行期间进行对象替换。

(3)模型的可移植性。因为模型是独立于视图的，所以可以把一个模型独立地移植到新的平台工作。需要做的只是在新平台上对视图和控制器进行新的修改。

(4)代码和设计的复用性：广泛采用可复用的组件，能够降低新项目的开发成本，通过对设计的复用，使得开发小组之间更易于沟通，设计的系统更易于理解。同时还可以基于此模型建立应用程序框架，而不仅仅是应用在设计界面中。

2.2 MVC模式的缺点

（1）增加了系统结构和实现的复杂性。对于简单的界面，严格遵循MVC，使模型、视图与控制器分离，会增加结构的复杂性，并可能产生过多的更新操作，降低运行效率。

(2)视图与控制器间过于紧密的连接。视图与控制器是相互分离，但确实联系紧密的部件，视图没有控制器的存在，其应用是很有限的，反之亦然，这样就妨碍了他们的独立重用。

(3)视图对模型数据的低效率访问。依据模型操作接口的不同，视图可能需要多次调用才能获得足够的显示数据。对未变化数据的不必要的频繁访问，也将损害操作性能。

2.3 MVC模式的应用及理解

在对MVC模式的研究并将其应用在项目开发的过程中，对MVC模式有了以下的几点认识和体会：

(1)MVC不仅是一种设计模式，更是一种抽象思想。它体现了合理抽象、封装和分离以实现结构清晰易于维护的原则。

(2)注意合理的使用MVC的范围。在软件系统开发中，并非所有的部分都要严格地遵循MVC模式，强行使模型、视图与控制器分离。因为有时候这种做法会增加结构的复杂性，同时过多产生的更新操作会降低系统运行效率。

(3)要充分理解MVC模式在实际运用中的状况，因为传统的MVC模式只是从概念上将视图从流程控制、业务逻辑中独立出来，并定义了相互间作用的机制，使各个模块的开发相对独立，但是它没有针对不同视图类型，解决流程控制等对象的统一问题。因此，针对具体的研究对象，我们在使用MVC模式时需要对此做进一步的研究。

结语：

体系结构的选择往往是一个系统设计成败的关键。本文对MVC模式进行了深入剖析，探讨了它的结构及优缺点，为成功地应用MVC模式提供了理论依据。MVC设计思想，使得逻辑处理和应用表示相分离，降低了模块之间的藕合度。它不仅可以应用在软件系统的设计分析阶段，同时还可以应用在软件系统的实现阶段。