MVC 模式在 B/S 结构政务系统的应用研究

任广震 侯 进* 王 献
(西南交通大学信息科学与技术学院 四川 成都 610031)

摘 要 随着信息技术的发展和政府工作方式创新的理性选择,电子政务应运而生,并成为联系政府和公众的一个重要窗口。通过体系结构的研究,提出使用基于 B/S 多层结构作为电子政务系统的主流架构体系。分析 MVC 设计模式的特点,研究 MVC 基于 ASP. NET 平台在 B/S 结构系统中的应用,并以某市开发区电子政务系统作为实例,对系统结构、各功能模块以及关键技术的实现进行了详细的描述。

关键词 电子政务 体系结构 B/S 结构 MVC ASP. NET

中图分类号 TP311 **文献标识码** A DOI:10.3969/j.issn.1000-386x.2014.08.014

ON APPLYING MVC MODE IN E-GOVERNMENT SYSTEM WITH B/S STRUCTURE

Ren Guangzhen Hou Jin* Wang Xian
(*School of Information Science and Technology, Southwest Jiaotong University, Chengdu 610031, Sichuan, China*)

Abstract E-government comes into being along with the development of informatisation technology and the rational choice of governmental works innovation, and becomes an important window connecting the government and the public. By studying the system structure, we propose to use B/S-based multi-layer structure as the mainstream architecture system of E-government system. We analyse the features of MVC design mode and study the application of MVC based on ASP. NET platform in B/S structure system. We use an E-government system in the development zone of a certain city as an example, and give a detailed description on the implementation of system structure, functional modules and key technologies.

Keywords E-government System structure B/S structure MVC ASP. NET

0 引 言

随着 Web 开发技术发展日趋成熟,目前电子政务系统的开发越来越多地采用基于浏览器的 B/S 系统架构,逐渐舍弃复杂的 C/S 结构。但在开发 B/S 应用的过程中,部分开发人员往往将业务逻辑和页面显示混合在一起,不利于分工与协作;而且在业务逻辑中采用内嵌 SQL 语句的方式完成数据访问,一旦数据库或者类定义中一方发生变化,就会导致系统的大幅修改,不利于系统的维护。针对以上不足,本文提出基于. NET 的 Web 多层架构体系,使用 MVC 设计模式应用于 B/S 的开发之中,并完成了在 ASP. NET 平台的设计与实现,为解决系统应用的不足提供了一种解决方案。

电子政务[1] 是把工业化模型的大政府转变为新型的服务性政府,以适应虚拟的、全球性的、以适应知识为基础的数字经济,同时也适应社会的根本性转变。怎样运用先进的技术构建电子政府实现电子政务一直是各国追求的目标和研究的重要课题;而且随着经济社会的不断发展,政府组织部门的不断变革,电子政务的应用过程中要迎合政务活动的多变性所带来了的挑战;因此电子政务系统必须具有更好的重用性和可扩展性也就成为电子政务系统开发中需要重视和解决的问题。通过对系统体系结构和 MVC 设计模式的研究,并将应用于电子政务系统的开发过程当中,很好地解决了电子政务系统应用的问题,使电子政务得到了更好地发展和利用。

1 系统体系结构

1.1 传统 C/S 体系架构

C/S 架构体系即客户机/服务器体系结构,出现于 20 世纪 80 年代,最早由美国 Borland 公司研发。C/S 架构是传统的两层结构如图 1 所示:第一层是客户机系统,结合了用户界面与业务逻辑,完成与用户的交互任务;第二层是服务器层,负责数据管理。C/S 架构属于胖客户机结构,通过客户应用程序直接访问服务器,运行速度较快,服务器负担较轻。但是随着现代网络技术的飞速发展、用户更高要求的提高,C/S 模式的应用扩展性低,软件移植、维护和升级困难的问题逐渐暴露出来。

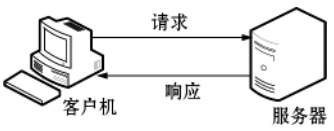

图 1 C/S 两层体系结构

收稿日期:2013 - 03 - 21。国家自然科学基金项目(60902023,61171096);四川省动漫研究中心科研项目(DM201204);中央高校基本科研业务费专项资金项目(SWJTU12CX093)。任广震,硕士生,主研领域:电子政务。侯进,副教授。王献,副教授。

1.2　B/S 三层架构

B/S 架构体系[2]即浏览器/服务器体系结构，发展于 20 世纪 90 年代，由美国微软公司研发。B/S 架构属于瘦客户机结构，是 3 层架构体系，如图 2 所示：第一层为客户端的浏览器层，是用户输入数据和显示结果的交互界面；第二层 Web 服务器层，主要负责对客户端应用程序的集中管理；第三层数据库服务器，主要负责数据存储和组织、数据库的分布式管理、数据库备份和同步等。

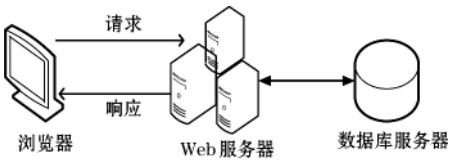

图 2　B/S 三层体系结构

用户在浏览器表单中输入数据，然后将表单中的数据提交并发送到中间服务器，中间服务器应用程序接受并处理用户的数据，并转化为与数据库的交互操作。最后中间服务器把返回的结果传送到客户端，在浏览器中显示出来，这样就大大降低了客户端负担，减轻了系统维护与升级的成本和工作量。

C/S 架构和 B/S 架构是计算机领域两种比较流行的体系结构，它们各自具有其自身的特点，现在对 C/S 架构和 B/S 架构进行详细地比较与分析，如表 1 所示。

表 1　C/S 架构和 B/S 架构的分析比较

	C/S 架构	B/S 架构
升级和维护性	困难	容易
客户端负载	大	小
灵活性	低	高
技术标准	内部统一	开放
用户界面风格	不尽相同	统一友好
支持用户量	较少	多
交互性	强	较弱
安全性	强	较弱
开发维护成本	高	较低
培训时间费用	高	低

由表 1 可知，B/S 架构较好的弥补了 C/S 架构的不足，实现了动态的、交互的、分布与跨平台的以及易于集成和扩展的操作。

1.3　基于. NET 的 Web 多层架构体系

随着越来越多的企业级分布式事务处理要求的提高，三层体系结构中 Web 服务器的工作负荷也在不断加大。为了解决以上问题，构建出高质量、高度可用的系统应用，需要继续对 Web 服务器的层次结构进行细分，然而对于细分的方法根据不同的技术平台而不同。本文针对. NET 平台，将 Web 中间服务器继续细分为表示层、业务逻辑层、数据访问层，从而形成的基于. NET 的 Web 多层体系结构如图 3 所示。

图 3　基于. NET 的 Web 多层体系结构

表示层(UI)：定义了用户可以在计算机屏幕上观察到的内容，包括用户界面和用户体验；业务逻辑层(BLL)：负责添加业务规则和领域相关的对象，与系统业务特有的逻辑相关，用于满足应用程序的业务需要。如果涉及到数据库的访问，则调用数据访问层；数据访问层(DAL)：负责获取和操作保存在数据存储中的持久化原始数据，并以一种更为抽象和直观的方式将数据存储暴露给应用程序同时需要隐藏数据存储的底层细节；数据存储层(DS)：这一层用于保存数据，可以使一个关系数据库，或者是一个 XML 文件或文本文件，还可以是其他类型的专有存储系统。应用程序可以在数据存储层中持久化数据。

1.4　基于. NET 的 Web 多层架构体系在电子政务系统应用的意义

基于. NET 的 Web 多层体系结构具有更好的易用性、较强的健壮性、高度的安全性等特点，充分满足了电子政务系统应用的要求。不仅如此，基于. NET 的 Web 多层体系结构以其独特的优势使政务系统在行使职能的过程中发挥了更加重要的作用。现在从以下几个方面进行具体分析：

1) 实用性更强、易于发布。基于. NET 的 Web 多层体系结构属于 B/S 结构，压缩了客户端的功能将客户端应用程序写得很小，而把大多数工作交给中间层处理。瘦客户端还有利于系统的发布，因为它们不需要再考虑安装、配置和维护数据库连接软件的问题。电子政务系统面对的大多是政府工作人员，这就决定了不能有过高的操作复杂性，通过浏览器就能很好进行业务操作，基于. NET 的 Web 多层体系结构正好满足了这种要求。

2) 更好的稳定性和可靠性。电子政务系统应用于政府的日常工作管理当中，有些环节关系到国计民生，这就需要其长时间保持稳定地运行。多层次结构的设计可以将最重要的数据信息作为单独的一层进行设计，可以保证在系统出现问题时，原有的重要数据不会丢失。而且其中一层的改变，对其它层的影响可以减小到最低。此外，多层体系结构将服务器端工作负荷和业务逻辑分布到不同层次上进行，不同层次的模块单元可以部署在不同机器上，这样就避免了个别机器工作负荷太重从而降低了系统出现问题的可能性，提高了系统的稳定性。

3) 可重用性强。随着社会经济的发展和组织部门的变革，政务活动具有多变性，电子政务系统总是在不断地扩充与变化，如何在整合系统中原有的功能模块基础上去完善和扩充新的功能，这就对电子政务系统可重用性和扩展性提出了更高的要求。基于. NET 的 Web 多层体系结构提供了完善的组件技术并且把业务逻辑封装在一层进行共享，这样可以更方便地整合原有的电子政务系统，从而更好地进行扩展。

4) 更高的安全性。电子政务系统行使政府职能，有些环节会涉及到国家重要信息，这就对电子政务的安全性提出了更高的要求。基于. NET 的 Web 多层体系结构可以通过使用不同的访问约束，来分层隔离敏感的功能。这就提供了一个灵活的和可配置的安全层。对不同层级分别进行安全设施，层层进行保护，这样就提高了电子政务系统的安全性。

综上所述，基于. NET 的 Web 多层体系结构具有高度易用性，较强的稳定性和可靠性，良好的可重用性和扩展性以及更高的安全性等优势，保证了政务系统高效的行使各种政府职能，因此将基于 B/S 多层体系结构作为电子政务系统的主流架构体系。

2 MVC 设计模式

2.1 MVC 的介绍

MVC 由 Trygve Reenskaug[6] 提出，是“Model-View-Controller”缩写，中文翻译为“模式-视图-控制器”。对于界面设计可变性的需求，MVC 把交互系统的组成分解成 Model（模型）、View（视图）、Controller（控制器）三种部件。

模型是 MVC 设计模式的核心，封装了系统的核心流程和业务规则，是软件所处理的问题逻辑在独立于外在显示内容和形式情况下的内在抽象，它独立于具体的界面表达和 I/O 操作。

视图是用户看到并与之交互的界面，主要负责向用户显示相关数据，并能接受用户的输入数据但是它并不进行任何实际的业务处理。它从模型获得显示信息，对于相同的信息可以有多个不同的显示形式或视图。

控制器是定义应用程序的行为，解释用户动作，负责模型和视图之间的同步，接受用户的输入并调用模型和视图去完成用户的请求。通常一个视图具有一个控制器。

对于每一个用户输入的请求，首先被控制器接收，并决定由哪个模型来进行处理，然后模型通过业务逻辑处理用户的请求并返回数据，最后控制器用相应的视图去格式化模型返回的数据，并通过显示页面呈现给用户。图 4 所示为模型、视图、控制器这 3 个模块各自的功能以及它们之间的相互关系。模型、视图与控制器的分离，使得一个模型可以具有多个显示视图。如果用户通过某个视图的控制器改变了模型的数据，所有其它依赖于这些数据的视图都应反映到这些变化。因此，无论何时发生了何种数据变化，控制器都会将变化通知所有的视图，导致显示的更新。这实际上是一种模型的变化-传播机制[7]。

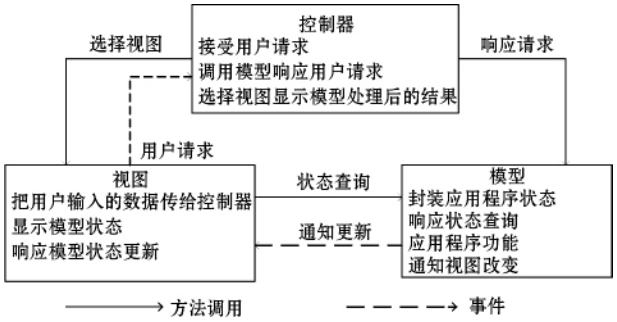

图 4 MVC 模式结构图

2.2 MVC 的优势

首先，一个模型能够建立和使用多个视图。变化-传播机制可以确保所有相关的视图及时得到模型数据变化，从而使所有关联的视图和控制器做到行为同步。

其次，模型层是自包含的并与视图、控制器相对独立，它们之间层次分明、结构清晰，这种松耦合的特性可以使开发人员和设计人员关注于特定层，有助于应用的并行开发，也为系统中后期的可扩展性、可维护性带来了极大的便利。

此外，MVC 中的控制器负责控制业务对象完成相应的请求，从模型层获取数据并指定相应的视图进行数据的呈现，甚至还可以在运行期间进行各个对象之间的替换，这种强大的控制手段大大提高了应用程序的灵活性和可配置性。

2.3 MVC 的不足

首先增加了系统结构的难度和实现的复杂性。对于简单的界面，如果严格按照 MVC 设计模式，使模型、视图与控制器分离，增加了结构的复杂性，并可能产生过多的系统交互，降低运行效率。

其次，MVC 理论上要求视图和控制器的分离，但实际中总会存在一定程度的联系。视图与控制器是相互分离的部件，但确实联系紧密，视图没有控制器的存在，其应用是很有限的，反之亦然，这样也会妨碍了他们的独立重用。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想要显示的数据。然而一些不必要的频繁访问，也会对系统的性能造成一定的损害。

最后，对 MVC 的准确理解和把握并不是那么容易。运用 MVC 需要精心的计划，由于它的内部原理比较复杂，所以需要花费一些时间去思考。

3 MVC 在 B/S 结构系统的应用的实现

ASP. NET 是微软. NET 框架的一部分，是一个统一的 Web 开发模型，是建立、部署及执行 Web 应用程序的平台，它为构造新一代动态网站和基于网络的分布式应用提供了强有力的支持。与以往基于过程的 ASP 页面技术相比，面向对象技术在 ASP. NET 中得到了完全实现。本文所讨论的 MVC 模式在 B/S 结构系统中的应用是在. NET 平台上实现的[8]。MVC 设计模式结合 ASP. NET 应用于 B/S 结构系统的设计实现如图 5 所示。

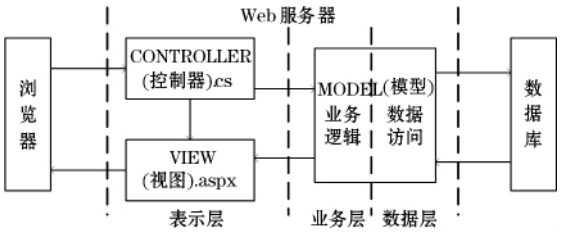

图 5 MVC 基于 ASP. NET 在 B/S 开发中的应用模型

MVC 设计模式基于 ASP. NET 平台在 B/S 开发中的应用模型如图 5 所示，结合 MVC 设计流程的步骤，如图 6 所示，对于系统的实现下面给出了详细地描述。

（1）系统应用的分析。分析应用问题，分离出系统的内核功能、对功能的控制输入、系统的输出行为三大部分。

（2）模型部件的设计。在 B/S 结构系统中，模型部件包括 Web 服务器三层结构中的业务层和数据层，它封装系统内核数据和计算功能，提供访问显示数据的操作，提供控制内部行为的操作以及其他必要的操作接口。这些部分在. NET 里可以写成独立的类文件，也可以打包成 DLL 文件，这样复用性将会更高。

（3）视图的设计。设计每个视图的显示形式，它从模型中获取数据，将它们显示在屏幕上。ASP. NET 中开发者在 . aspx 页面中设计用户界面来实现视图。

（4）控制器的设计。MVC 设计模式中的视图和控制器位于 B/S 三层架构中的表示层，控制器的作用是接收请求并决定调用哪个模型构件去处理请求，然后确定用哪个视图来显示模型处理返回的数据。控制器的设计可以在 ASP. NET 的逻辑控制文件. cs 里来实现，并且 ASP. NET 内部也提供了一套良好的控制部件。所以，控制器可以由. cs 逻辑文件和. NET 内部配置来实现。

（5）实现独立的彼此分离的控制器。控制器的独立性，带来了更高的自由度，并且帮助形成高度灵活性的应用。在 ASP. NET 中，每个. aspx 页面都会有一个. cs 文件与之对应，不同页面的彼此独立也使. cs 文件之间相互独立，这样就满足了

控制层的一个条件;然而，控制层与视图层总是存在一定程度的耦合关系而不能达到严格意义上的分离。MVC 设计模式的应用并不是一味地全盘照搬，而是有选择地使之适应这个工程,这样才能更好地将 MVC 思想融入到系统设计之中。在 .NET中页面与. cs 文件虽然关系紧密,不过这样也使页面响应与反馈更加专一。就像 Observer 设计模式[7] 中“观察者”所起到的作用一样，一旦观察者都得到通知,作为对这个通知的响应，每个观察者都将查询目标以使其状态与目标的状态同步。这种方式更能符合查询的系统要求。所以我们要做的是尽可能地降低控制层与视图的耦合。解决办法是在 . cs 文件中只编写控制逻辑,而有关视图方面的尽量都写在页面中,也可以使用 HTML 语言或脚本语言如 JavaScript 进行辅助。

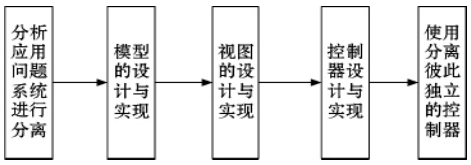

图 6　MVC 模式设计流程

4　某市开发区政务系统的实例分析

4.1　需求分析与结构设计

随着某市经济开发区的不断发展,开发区企业的不断增多,开发区的管理和业务处理量越来越大,怎么样保证开发区与企业之间信息的高效准确地传达,及时地了解开发区项目建设进度以及人才市场的情况,保证开发区管委会和各企业有效地沟通,更好地服务于开发区的发展,是我们十分重视的问题之一。

开发区管理委员会为市政府的派出机构,受市政府委托行使市政府有关权限,负责经济开发区的建设和管理。管委会内部共分为办公室、投诉中心、建设环保分局、招商分局、财政分局等十个部门。管委会各行政部门负责不同的管理业务,不同部门之间既明确分工又相互合作。但是随着管委会业务量的增加,管委会机构设置的增加,怎么样更好地处理业务流转过程,转变原有工作方式,切实提高工作效率,这是电子政务系统设计中需要考虑的问题。

总之,开发区电子政务系统要利用先进技术整合各种资源,创建一个集成的高性能的办公环境,开发区管委会各领导、各局、室、中队、投诉中心工作人员可以随时随地开展行政管理各项工作,保障开发区各工作高效、安全运行,为行政管理本身的重整和发展带来新的机会。开发区电子政务系统包括政府门户网站和政务内部办公系统两个部分,政府门户网站对外用于信息发布与交流,为招商引资提供一个良好的平台;政务内部办公系统,为内部工作人员提供集成的办公环境和共享的工作平台,转变工作方式,提高工作效率。系统的功能结构如图 7 所示。

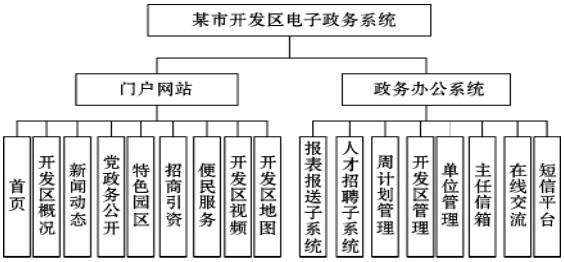

图 7　电子政务系统功能结构图

开发区电子政务系统采用 B/S 多层架构体系,系统的技术架构如图 8 所示。

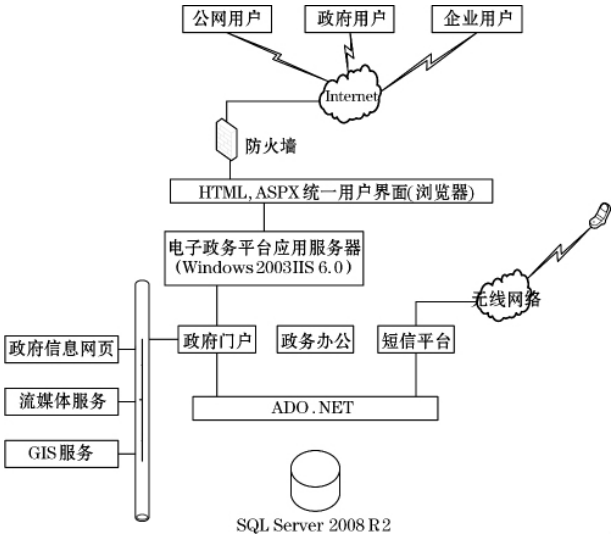

图 8　电子政务系统技术架构图

4.2　视图层的实现

本系统视图层页面的实现采用复合视图的形式,即一个页面由多个子视图(用户部件)组成;. NET 视图设计器提供了数据、Web 窗体、组件、HTML 等各种控件,同时还提供了用户控件。子视图可以是最简单 HTML 控件、服务器控件或多个控件嵌套构而成的 Web 自定义控件。

(1) 采用母板页,定义 css 样式等方法,达到界面风格和控件样式的统一,并采用超级文本标记语言修改页面。

(2) 对于界面中输入的不同的类型数据需要判断其正确性,采用正则表达式实现验证。

(3) 采用 Javascript 脚本语言,实现界面局部刷新,提高响应速度;另外还用 Javascript 处理一些较复杂的请求输入验证,增加系统的人性化操作。

本系统中视图与各模块中的. aspx 文件相对应,. aspx 文件与系统中的各个用户界面相对应,并以登录模块为例,系统登录模块页面如图 9 所示。

图 9　政务系统登录模块页面

4.3　控制层的实现

在开发区政务系统中,控制层的设计采用分散控制器的原则,每个. aspx 文件对应了一个后台代码. cs,在每个. cs 文件中来实现每个页面的页面控制器的功能。页面控制器捕获每个视图页面中发生的事件和提交的数据，并调用相应模型来处理它们,最后决定视图中的展现内容和方式。下面以系统常用的登录模块为例,在登录模块视图 Login. aspx 对应的 Login. aspx. cs 中实现 controller 的功能,根据一定的控制逻辑判断用户输入是

否为开发区用户，企业用户以及其他用户等，并根据用户的合法性以及合法用户的不同身份实现不同的登录效果，并对登录用户信息进行管理，具体实现如以下代码所示：

```
protected void Page_Load(object sender, EventArgs e)
        {
            if (! Page. IsPostBack)
              {
                Session. Abandon();
                Session. Clear();
                //清除 session
                username. Focus();
              }
protected void ImgLogin_Click(object sender, ImageClickEventArgs e)
{
                string strUserName;
                string strPassword;
                …
    if (rdoGov. Checked)
    //当用户类型为管委会用户时
    {SQLGov objSQLGov = new SQLGov();
    //根据用户名和密码获取管委会用户数据信息 tbl = objSQLGov.
GetAGovUserInfoForLoginTbl(strUserName, strPassword);
        }
        ……
        UserManager. SetUser(objCurUser);
        //进行用户信息管理
        Response. Redirect("frame. aspx");
        //跳转到指定页面
        ……
    }
```

4.4 模型层的实现

分析电子政务系统的内在属性，使其能够包含应用程序的数据以及对数据进行访问、修改及相关的业务逻辑规则，能表达出程序所使用的数据和应用程序的运行状态. 因此模型层涵盖了业务逻辑层(BLL)和数据访问层(DAL)。

(1) BLL 的设计与实现

BLL 是对业务逻辑的实现，它从 Controller 接收请求，执行业务逻辑处理，并将处理结果返回给 Controller，以供 View 显示，而 BLL 的业务处理一般直接调用 DAL 的某个方法来进行[10]，如以下代码所示。

```
    private void SaveUser()
    {
    SQLCorp objSQLCor = new SQLCorp();
    CorUser objCorUser = new CorUser();
        ……
objCorUser. ID = SQLCommon. GetSeqNumber("5");
iCnt = objSQLCor. AddCorUserInfo(objCorUser);
        ……
    }
```

如以上代码所示，在开发区电子政务系统中定义了一组实体类，每一个实体类对应数据库中的一个表或者视图，如 CorUser. cs 对应于企业用户信息表，GovUser. cs 对应于政府用户信息表等，用于接收调用 DAL 处理方法后返回的数据，该类只包括字段和对应于所有字段的属性，具体代码如下所示：

```
        public class CorUser
        {
            private string CorUserID;
            private string CorUserName;
            ……
            public string CorUser_ID
            {
              set { CorUserID = value; }
              get { return CorUserID; }
            }
    ……
    }
```

而数据集则有 DataSet 类定义的对象接收，最后 BLL 将这些数据返回给 Controller，以确定是刷新当前页面还是跳转到其他页面。这样就可以把数据库和政务系统业务对象进行有机的结合。此外，还将开发区政务系统中一些公共的业务逻辑、处理方法以及一些公用服务模块封装成单独的类文件，例如 Report-Manager. cs 是属于报表报送系统中的报表管理类，这样表示的业务逻辑非常清楚，源代码也易于维护。

(2) DAL 的设计与实现

在开发区电子政务系统中定义了核心类 DBhelperSQL. cs 用于执行系统中所有对数据库的访问，其它的数据库操作类如 SQLArticle. cs、SQLConc. cs、SQLWeekPlan. cs 等都依赖于 DB-helperSQL. cs 用于实现对数据库中文章信息表、在线交流表、周计划表等不同数据库表的各种具体操作。DAL 的设计如图 10 所示。

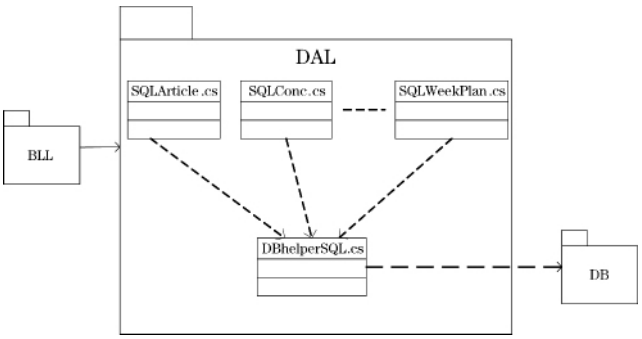

图 10 DAL 设计图

核心类 DBhelperSQL. cs 的设计如图 11 所示。

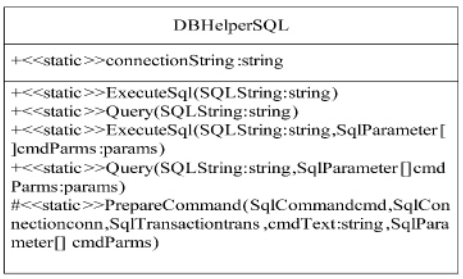

图 11 数据库操作类设计图

5 结 语

将 MVC 设计模式应用于基于 B/S 多层结构的系统之中，并在 ASP. NET 中为系统的设计与实现提供了一种很好的解决方案。实现了软件开发的分工和应用程序的模块化，提高了代码复用率，降低了维护成本。在满足用户需求的基础上，是一个

(下转第 74 页)

领域本体间关系的凝聚层次聚类算法进行改造，因而把聚类效果和本体效果结合在一起对领域本体生成算法进行评价。使用 F-测量值(F-Measure)[16] 对聚类结果评价，效果比较如表 2 所示，由此可见基于 K-means 的层次聚类方法在运行效率和聚类结果方面都取得了较大改进。

表 2 试验结果比较

结果比较	凝聚层次凝聚法	基于 K-means 的层次聚类法
运行时间(s)	52.893	16.342
F-测量值	0.685	0.823

4 农业领域本体的应用实施

将以上研究的农业领域本体构建方法应用到农业信息检索中，本文构建了一个基于本体的农业搜索引擎平台，运用垂直搜索引擎原理，对传统专业搜索引擎的结构进行改造，将本体作为网页分析过滤和关键字相关性判断的标准，得到更为高效的体系结构。如图 4 所示为平台的某个搜索界面，该平台集成了 Web 网络信息的抽取、本体的构建和基于本体的农业搜索等功能。

图 4 基于本体的农业搜索引擎界面

5 结 语

农业领域本体的构建为农业搜索引擎提供知识组织基础，提高了农业搜索引擎的性能。本文基于农业文献，把数据抽取、关联分析、聚类等知识发现技术加入到农业领域本体的构建过程中，从而提高农业领域本体的构建自动性。本文从 Web 网络抽取源数据，进行分词和清洗之后得到领域概念，并用关联分析和改进的层次聚类方法发现领域概念间关系，最后对构建的农业领域本体应用实施，构建了一个基于本体的农业搜索引擎。但是本文对构建的农业领域本体相关评价方法不精确，今后可以考虑在领域本体构建方法评价方面和农业领域本体应用方面进行更深入研究。

参 考 文 献

[1] Studer R, Benjamins V R, Fensel D. Knowledge Engineering, Principles and Methods [J]. Data and Knowledge Engineering, 1998, 25(1-2): 167-197.

[2] Fan J, Gao Y, Luo H. Integrating concept ontology and multitask learning to achieve more effective classifier training for multilevel image annotation [J]. IEEE Transactions on Image Processing, 2008, 17(3): 407-426.

[3] Wang H, Jiang X, Chia L T, et al. Ontology enhanced web image retrieval: aided by wikipedia & spreading activation theory [C] //Multimedia Information Retrieval, 2008: 195-201.

[4] Zheng H T, Kang B Y, Kim H G. An ontology-based approach to learnable focused crawling [J]. Information Science, 2008, 178(23): 4512-4522.

[5] Lai L F. A knowledge engineering approach to knowledge management [J]. Information Science, 2007, 177(19): 4072-4094.

[6] 金鑫. 面向 Web 信息资源的领域本体模型自动构建机制的研究 [J]. 计算机科学, 2012, 19(6): 213-216.

[7] Zhong N, Yao Y, Kakenoto Y. Automatic Construction of Ontology from Text DataBases [J]. Data Mining, 2011, 2: 173-180.

[8] Tao D, Li Yuefeng, Zhong Ning. A Personalized Ontology Model for Web Information Gathering [J]. IEEE Transactions on Knowledge & Data Engineering, 2011, 23(4): 496-511.

[9] Maeduche A, Staab S. Ontology Learning for the Semantic Web [J]. IEEE Intelligent Systems, 2011, 16(2): 72-79.

[10] 张磊, 李秀峰. 基于叙词表和文献数据库的农业领域本体构建方法研究 [D]. 北京: 中国农业科学院, 2011.

[11] 徐力斌, 刘宗田, 周文, 等. 基于 WordNet 和自然语言处理技术的半自动领域本体构建 [J]. 计算机科学, 2007, 34(6): 219-222.

[12] 刘军, 张净. 基于 DOM 的网页主题信息的抽取 [J]. 计算机应用与软件, 2010, 27(5): 188-190.

[13] 王曰芬, 宋爽, 卢宁, 等. 共现分析在文本知识挖掘中的应用研究 [J]. 中国图书馆学报, 2007(2): 59-64.

[14] 岑咏华, 王晓蓉, 吉雍慧. 一种基于改进 K-means 的文档聚类算法的实现研究 [J]. 现代图书情报技术, 2008(12): 73-79.

[15] 尉景辉, 何丕廉, 孙越恒. 基于 K-means 的文本层次聚类算法研究 [J]. 计算机应用, 2005(10): 2323-2324.

[16] 张磊, 李秀峰. 基于叙词表和文献数据库的农业领域本体构建方法研究 [D]. 北京: 中国农业科学院, 2011.

(上接第 58 页)

具有良好安全性、可扩展性和易维护的交互式系统。此外，所开发的实例对今后电子政务系统应用研究也有一定的参考价值。

参 考 文 献

[1] 王琦, 陈霞, 陈飞. 电子政务 [M]. 北京: 电子工业出版社, 2011.

[2] 李园, 陈世平. MVC 设计模式在 ASP.NET 平台中的应用 [J]. 计算机工程与设计, 2009, 30(13): 3180-3185.

[3] 刘亮, 霍剑青, 郭玉刚, 等. 基于 MVC 的通用型模式的设计与实现 [J]. 中国科学技术大学学报, 2010, 40(6): 635-639.

[4] 刘亚鹏, 张征, 俞婷. 基于 MVC 多层架构的 Web 应用框架设计 [J]. 微计算机信息, 2011, 27(7): 169-171.

[5] 郭晓峰, 姚世军, 尹祖伟. 基于.NET 的 Web 应用框架的设计与应用 [J]. 计算机工程与设计, 2008, 29(2): 454-459.

[6] 朱爱红, 余冬梅, 张聚礼. 基于 B/S 软件体系结构的研究 [J]. 计算机工程与设计, 2005, 26(5): 1164-1166.

[7] 赖英旭, 刘增辉, 李毛毛. MVC 模式在 B/S 系统开发中的应用研究 [J]. 微计算机信息, 2006, 22(10): 62-65.

[8] 黎永良, 崔杜武. MVC 设计模式的改进与应用 [J]. 计算机工程, 2005, 31(9): 96-98.

[9] 张金波. .NET 平台分层架构开发中泛型实现通用数据访问层 [J]. 计算机与数字工程, 2012, 40(11): 73-75.

[10] 孙健伟, 高岭, 王晓晔. .NET 开发中 MVC 模式的应用 [J]. 计算机技术与发展, 2007, 17(11): 8-11.