

基于 ASP.net 的工作流技术在 OA 系统中的应用

欧阳李彦

(重庆工学院 重庆 400050)

摘要 ASP.net 技术是 Microsoft.net 平台的重要组成部分,这种新的 Web 应用程序开发技术具有许多优点,基于 ASP.net 技术,采用合理的开发设计模型就能够得到结构合理、易于维护的 Web 应用程序解决方案。本文基于工作流的开发方法,用 ASP.net 技术,结合实际介绍了高校办公自动化系统(OA)的开发应用,并探讨了有关的技术问题,提出了一种用于 Web 应用程序开发的七层设计模型,从而降低了开发难度,提高了工作效率。

关键词 办公自动化,工作流,Asp.net

The Application of Workflow Technology in OA System Based on ASP.net

OU Yang LI Yan

(Chongqing Technology Institute, Chongqing 400050)

Abstract In the office automation system the workflow is always the important part. The paper brings forward seven-tier Architecture based on .NET framework and adopted in the office automation system, discusses the function fulfilled by every tier, analyzes the disposal model of workflow in the office automation system, and works over the definition of flow node. An example of design of the office automation with ASP.net is shown in the paper.

Keywords Office automation system, Workflow, ASP.net, Model, Design and implementation

1 ASP.net 的技术特点

(1)ASP.net 是一种基于 Microsoft.net 平台的 Web 应用程序开发技术,构建在公共语言运行时 CLR(Common Language Runt.)之上,可以使用 .net Framework 所提供的全部功能。用它开发的程序可以支持异常控制、类型安全、继承和动态编译。

(2)ASP.net 程序中还可以使用多种支持面向对象编程的强类型语言编写控制逻辑,大大提高了其执行效率和跨语言的兼容性。

(3)ASP.net 使用事件驱动与数据绑定开发方式,将程序代码与用户接口分开,使程序具有良好的可读性。

(4)ASP.net 使用 Sessionstate 管理方式,状态的信息可在不同机器的不同进程上维护,有效地解决了一个服务器上多个进程中对 Session 的管理。

(5)ASP.net 运行时密切监视和管理进程,若发生了错误行为,如漏洞和死锁,新的进程会在当前的位置建立,这样服务器可以不用重新启动从而保证服务器的连续运行。

(6)ASP.net 代码属编译方式执行,ASP.net 页被第一次请求时就编译进 .net 类中,且编译的类被缓存,用于随后的请求,因而以后每次执行无需加载,其执行性能和速度都有较大的改进和提高。

所有这些特点使得 ASP.net 技术可以完全采用面向对象的设计思路和编程方法来高效地开发 Web 应用程序,针对 Web 应用程序自身的特点,采用优化合理的设计和开发模型,运用 ASP.net 技术所提供的技术优势,开发出优秀的 Web 应用程序。

2 OA 工作流的处理

OA 系统中最关键的问题是如何对工作流进行处理,目前对工作流分析的理论和实践都不少。高校和政府机关中的 OA 系统的用户大多对计算机操作不够熟练,因此,对流的简洁处理是 OA 系统中要解决的关键问题之一。通过对高校实际需求的分析,将工作流分成以下两大类:静态流和动态流。将工作流上的每一个处理点称为流的结点。静态流是指流程固定的、有严格制度规定的工作流,此类工作流的流结点是固定不变的;动态流是指在流的运行过程中才能根据流的实际情况动态的确定结点的流。相比之下,静态流只是业务逻辑已经确定的程序,没有过多的研究价值,这里主要分析动态流的处理方法。

工作流是由流结点和结点之间的运动方向所构成,可以表示为: $S(V, E)$, V 是 S 的结点集, $E = \{(v_i, v_j) | v_i, v_j \in V\}$ 。如图 1 所示。

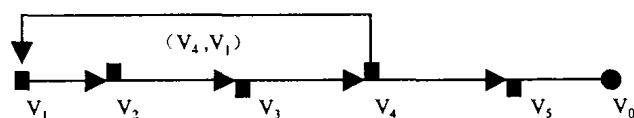


图 1 OA 中工作流示意图

从图 1 中可以看出,工作流的关键在于定义流的结点。

在 OA 应用系统中,考虑到用户的计算机使用水平,不能够将工作流的定义操作做得过于复杂,但又要能够充分满足 OA 中各种复杂的流的定义,我们采用如下方式进行:

• 建立流结点。分析实际中的各种工作结点,将这些结点进行抽象归纳,建立一个工作结点的基类,根据各部门的工作

性质不同,将各部门的结点从基类中继承。工作流是由结点完成的,每个结点都有一个绑定的“结点处理人”,“结点处理人”既可以是“单位”,也可是“个人”,如果处理人是“单位”,则要转换成相应的个人。系统中对每个单位设置了“收文负责人”和“发文负责人”两种用户,以便在发文阶段和收文阶段将“单位”结点转换成“个人”结点,通过这种转换,将面向单位事务和面向个人的事务都转换为特定的“个人”用户了,这样为将OA系统集成到“数字化校园系统”奠定了良好的基础。

• 结点的类型与权限。根据公文处理的不同阶段,归纳出各个阶段的结点类型,如“审核结点”、“签发结点”、“发文结点”等。在这些结点的属性中已经描述了结点的权限,结点与“结点处理人”绑定后,“结点处理人”的权限便就动态绑定了。这种处理方法与大多数OA不一样,传统的OA中是先定义用户的权限,根据不同权限,用户有不同的功能;我们的做法是,动态定义结点的处理内容,用户在处理此结点时,可以处理结点中规定的内容。本文中这种权限分配的方法称之为“结点权限”法。

• 静态定义流。“流”需要一个初始化的过程,因此系统中可以指定一部分用户拥有“发文权限”,这类用户可以建立“流”。流的建立者根据实际的需要,静态定义流的结点。静态流的实质是“链表”,故称静态流为“链流”,“链流”的结点是串行的(如图1所示)。同时定义多个流,就形成并行的结点了,如图2所示,如果把一个“链流”看成是一颗特殊的“树”,那么这种并行流的实质是“森林”。

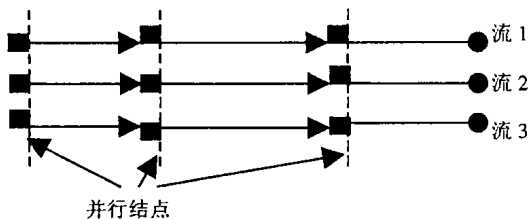


图2 并行流形成并行节点

• 动态定义流。如果静态流的某个结点在处理时,需要定义另一个工作流来完成本结点的处理,就需要使用动态定义流。如需要某单位签阅一份文件,传到这个单位的“收文负责人”,“收文负责人”需要定义一个流,以使单位的三位领导都来签阅这份文件。动态定义流的设立可以满足各类复杂流的定义。其图示如图3所示,动态定义流实质是“树”,因此动态流又称之为“树流”。

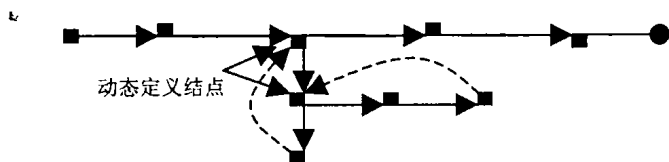


图3 OA中工作流示意图

3 OA系统的分层设计

OA系统采用分布式组件的方式不仅大大提高了系统的稳定性和安全性,同时也大大降低了硬件设备的成本。

传统的三层结构模式是:客户端、应用服务、数据服务三层。随着应用系统的复杂和庞大,传统的三层结构需要进行一定的扩充,以增加更大的灵活度来满足与应用系统相适应的软件工程方法。我们在设计OA系统构架时,将系统分成七

层:客户端、接口层、远程服务层、业务规则层、业务逻辑层、数据访问层、数据库层。实际上是将传统三层结构中的应用服务层扩展了五个相应层次,这是为适应网络分布式计算进行的必要扩充。

• 客户端层根据实际需要分为两种形式:定制的客户软件 and 通过的Web浏览器。采用定制客户软件的原因有两种,第一:提高系统的安全性;第二:提高用户的工作效率。对于.net技术而言,浏览器中不再有ActiveX的概念,统一使用可即时下载到客户端的Assembly(装配)来替代ActiveX,这种方式实际与使用的定制客户软件无任何实质性的区别。

• 接口层。由于客户端和组件服务器并未处于同一进程之中,其间的联系需要靠Agent来实现,接口层就起到Agent的作用,作为客户端而言,接口层是服务器的Agent;作为服务端而言,接口层是客户端的Agent。在.net中,典型的接口层写法示例如下:

```
public interface IUpdateModuleServer
{
    DataSet GetClientVersion();
    DataSet GetFileInfoUpdate(string moduleNo);
}
```

• 远程服务层。在.net技术中有多种方式可以实现分布式计算,常用的有两种:XML Web Service和Remoting。从本质上而言,这两种方式并无区别,但前者只能使用http协议^[1],可以方便地穿越防火墙(使用的是80端口);而后者既可使用http协议,也可使用TCP协议,但在防火墙上需要打开相应的服务端口,但对于同一局域网的服务器群而言,使用TCP的效率要高一些。无论采用哪种方式,都需要有一个统一的服务界面,远程服务层就是起到服务界面的作用。

• 业务规则层。由于客户端可以通过接口直接调用服务器端的服务,这样就带来了服务的身份认证和安全性问题。需要在服务界面的后面设立一些服务规则,只有符合规则的请求才能正确服务。业务规则层的主要任务是完成规则的检验。业务规则层还负有捕捉服务中的出错问题,当服务程序中出现了错误,业务规则层要能捕捉,并且用安全的形式返回相应的值,保证服务不会被中断。

• 业务逻辑层。系统中进行计算和操作的所有逻辑规则全部封装在这一层中,如SQL语句,OA中各种算法的实现等等。在日后系统的维护中,主要是维护这一层的程序代码,如:实际应用中的需求发生了变化,只需要将此层中的相关代码进行更改即可,不需要更改其它层的内容。

• 数据访问层。此层负责用户程序与后台数据库进行数据交换。如:根据用户要求需更换后台数据库系统,则只需要更改此层中的相关代码。

• 数据库层。管理所有的系统数据。

4 OA系统的工程实例

依据以上理论,我们在极短的时间内应用ASP.net技术开发了我校的OA系统——cqitOA系统。cqitOA系统是我校数字化校园的一个重要的应用系统,它建立在数字化校园的基础框架之上,如用户身份认证、电子邮件、FTP、计费网关等。

(1)框架的构建 cqitOA系统共分为7层框架,其中客户端层分为Client模式和Web模式两种。Client模式主要解决公文的建立和传递,Web模式进行公文的发布和查阅。数据库层采用Solaris+Oracle 9i。中间五层共同完成应用服务

的工作,这是从纵向的角度来看的;如果从横向的角度来看,cqitOA 为了平衡计算负荷量,将应用服务分为了 4 个子模块,分别为登录服务(也是数字校园框架的基础模块)、OA 事务逻辑服务(负责 OA 的事务流处理)、文件存取服务(负责将文件二进制内容写入数据库或从数据库中读取)、短消息处理服务(负责收发、解析短消息内容)。

从 cqitOA 系统结构的纵向来,共有 7 层服务;从横向来,由 4 个子模块构成,这 4 个子模块从物理位置上可以放置在校园网的任何位置,这样可以尽可能灵活地满足高容量、高效率的办公需求。图 4 是 cqitOA 的系统结构示意图。

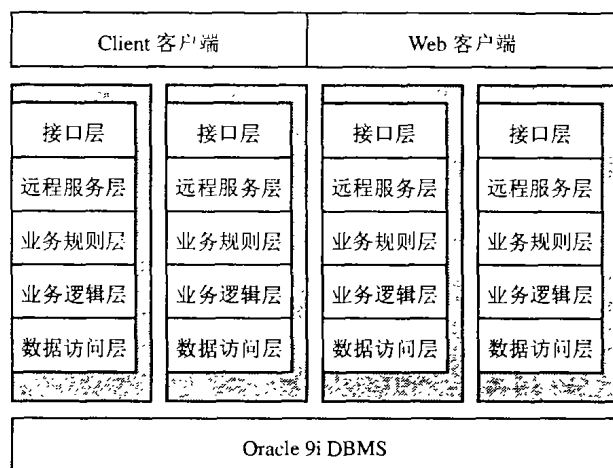


图 4 cqitOA 系统框架

(2) 权限管理 cqitOA 是依托于而且是服务于数字化校园框架的,校园网的教师、学生用户同时也就是 cqitOA 系统的用户,这些用户都必须是真实署名的,因此权限管理可以使用“结点权限”法。在系统中,对于个人用户而言只有两种权限:发文权限和申请发文权限。对于单位也设立了两种权限:发红头文件权限和发普通文件权限。如果单位具有发红头文件的权限,则该单位具有发文权限的用户则具备发红头文件的权限,通过这种转换方式,便可得出每位用户的各种发文权限。具有发文权限的用户可定义“发文事务流”,在定义的流结点时,同时定义了该结点的处理内容,应用“结点权限”法,处理该结点的用户也被赋予了相应的处理权限。

(3) 事务流的定义与处理 cqitOA 系统中事务流分为静态流和动态流两大类。最典型的静态流用于红头文件的建立过程。红头文件的建立过程国家有严格规定的标准程序,因此在此过程中的每一个结点所处理的事务是固定不变的,只是根据需要可以减少某些步骤,但并不能更改步骤的顺序。如:起草、审核、签发、复核、发文这些功能结点的顺序是不能变化的。

cqitOA 系统中动态流的应用就比较广泛了,最典型的应用是“申请类公文”的传递过程。如:某部门需要向校领导或相关部门申请一笔购买设备的资金。这样的功能流就是完成动态建立的,并不能事先规定好其功能结点,每一个结点完成后才能决定下一个结点该由哪个用户处理,处理什么内容。

流的可视化定义和查询是流设计中的一个难点,cqitOA 系统中有一个既简单又直观的处理方法。前面已经分析,无论是静态流还是动态流,都可以将“流”看成是“树”,而并行运行的流则组成“森林”。按照这个思路,我们利用“树型控件(TreeView)”来直观地表现“流”。.net 中的 TreeView 可以看

成是“森林”的容器,TreeView 中的每个 Node 则可以看成是结点。每个 Node 可以递归的定义其下级 Node,因此每个 Node 都可以形成一颗“树”,这种由 Node 组成的“树”完成可以系统动态的构建。TreeView 控件使用简单,又能够直观地反应“树”型的“流”,因此采用 TreeView 控件来表现和控制 OA 系统中“流”具有独到的优势。实际应用也证明,TreeView 控件也能够完成满足 OA 中各种复杂事务流的表现与控制,反过来证明了我们提出的“树流”的概念。图 5 是 cqitOA 系统中常见的“流”定义和查看界面。

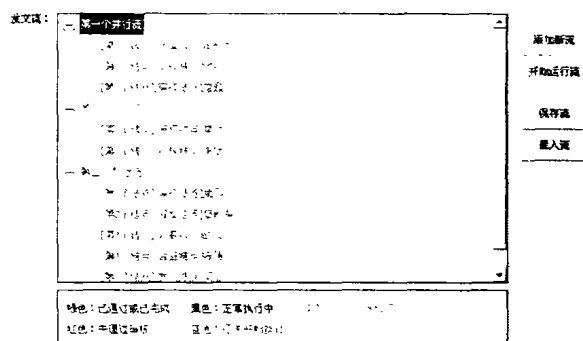


图 5 cqitOA 系统中流的处理界面

5 新技术的应用

考虑到学校和政府的 OA 系统用户大多是对计算机操作并不熟练的,在 OA 系统中引入一些新的技术元素可以极大地提高用户的可用性,还会增加用户对使用 OA 系统的兴趣。

• 手写技术。如果要求在批复公文时用键盘打字或用其它的方式进行文字录入,用户会觉得效率太低,或是感觉不习惯,这常常是 OA 系统难以在单位内部推广的主要原因。我们在 OA 系统中自主开发了一套可以直接手写批复、手写签名的子系统,其原理是利用 .net 框架中的 System.Drawing 和 System.Drawing.Imaging 两个 namespace 中的类记录鼠标的移动轨迹形成相应的 bitmap 文件。这项技术在实际应用中收到良好的用户反响。

• 手机短消息技术。有时候 OA 系统的用户在外出差,但又有文件需要审阅,就可以使用手机短消息技术,将 PC 操作平台扩展到普通的手机上。用户可以即时收到公文处理通知的短消息,也可以直接在手机上回复短消息以达到回复公文的目的。

参考文献

- 1 ASP.net Web Forms. MSDN Library For Visual Studio. net. Microsoft Corporation. 2001
- 2 Chappell D. Understanding .NET: A Tutorial and Analysis. 美国: Pearson Education, 2003
- 3 王超,张鹏. ASP/XML 深入编程技术[M]. 北京:北京希望电子出版社,2001
- 4 胡迎松,彭利文,池楚兵. 基于 .net 的 Web 应用三层结构设计技术. 计算机工程, 2003,29(8):173~175
- 5 任开银,黄东. 在 .net 上架构企业级应用程序. 微型机与应用, 2003(1):29~31
- 6 Thai T, Lam H Q. NET Framework Essentials. 美国: O'Reilly, 2003