第16卷 第7期
2006年7月

计算机技术与发展
COMPUTER TECHNOLOGY AND DEVELOPMENT

Vol. 16 No. 7
Jul. 2006

基于ASP.NET的Web应用系统架构探讨

王 蕾,李培峰,杨季文
(苏州大学 计算机科学与技术学院,江苏 苏州 215006)

摘 要:提出了一种基于ASP.NET开发方式的四层架构的Web应用系统构造思想。其基本内容是:将面向对象的UML建模与Web应用系统开发相结合,将整个系统分成适合ASP.NET开发方式的应用表示层、业务逻辑层、数据访问层和数据存储层四层结构。以此方式构造的Web应用不仅达到了代码组织结构清晰明朗、高重用性、适用性,易于维护和移植的目标,而且可以提高Web应用系统的开发速度。解决了目前大型Web程序开发中,代码适用性、重用性差,及难于维护和移植的问题。

关键词:Web应用系统;四层架构;ASP.NET
中图分类号:TP311 **文献标识码**:A **文章编号**:1673-629X(2006)07-0055-02

The Web Applied System Structure Based on ASP.NET

WANG Lei, LI Pei-feng, YANG Ji-wen
(School of Computer Science and Technology, Suzhou University, Suzhou 215006, China)

Abstract: Put forward a thought of the four layers structure of the Web applied system structure which based on the ASP.NET exploiture method. Its basic content is: combined the Object-Oriented UML modeling with the Web applied system exploiture, divided the whole system into four layers which suit for the ASP.NET exploiture method: application expression layer, operation logic layer, data interview layer and data storage layer. By this method to construct the Web applications not only makes the code organization structure clear and lucid, high-reusing, applicability, easy to maintenance and transplant, but also raises the development speed of the Web application system. It resolved the problems about currently the large Web procedure development, which are low-reusing of code applicability and difficult in maintenance and transplant.

Key words: Web applied system; four layers; ASP.NET

0 引 言

随着计算机网络技术的迅速发展,Web应用系统越发变得日益广泛起来。ASP.NET是微软推出的新一代Web开发平台,与其它Web开发技术相比,ASP.NET提供的Web页面级状态管理功能、服务器控件触发事件的工作模式、代码和内容分离的编程方式等[1],在一定程度上改变了以往的Web应用系统的架构模式。在软件开发技术方面,面向对象技术和软件分层结构设计是代码组织的一些好方法。但是对于具体的开发平台而言,多层结构有着不同的具体表现;对于具体的项目开发而言,面向对象技术对具体问题进行类定义和对象划分也不尽相同。因此,如何基于ASP.NET这些新技术在ASP.NET平台上应用面向对象技术架构一个逻辑清晰、模块合理的多层结构的Web应用系统就成了文中讨论的内容。

下面以笔者曾参与的《实验室信息管理系统》中的"论文管理子系统"为例,来阐述基于四层架构的设计思想。

收稿日期:2005-11-08
作者简介:王 蕾(1980-),女,河南开封人,硕士研究生,主要从事中文信息处理;杨季文,副院长,教授,主要从事中文信息处理。

1 ASP.NET下Web程序架构

系统描述:学生可以登记自己的论文,也可以删除自己登记的论文和相关所有回复;老师和学生可以查看、回复论文,也可以删除、修改自己的回复;管理员可以查看、删除登记的所有论文和回复。

笔者省略了论文管理系统的使用案例图和使用案例事件流图[2,3],只给出了修改自己回复案例的时序图(见图1)。

对于基于ASP.NET的Web应用系统,用户直接面对的是客户端浏览器,用户在使用系统时的请求是通过HTTP协议传递给服务器端的ASPX页面,用户操作的事务逻辑处理和数据的逻辑运算由服务器与数据库系统共同完成。按照在系统中的用途分类,把负责系统与角色交互的对象称为边界类对象,把负责系统中访问数据库的对象称为实体对象,把系统中介于边界对象和实体对象之间,负责时序图中流程的对象称为控制对象。故在时序图(见图1)中,Papers.ASPX和Session对象属于边界类对象,Users对象属于控制类对象,Reply对象属于实体类对象。

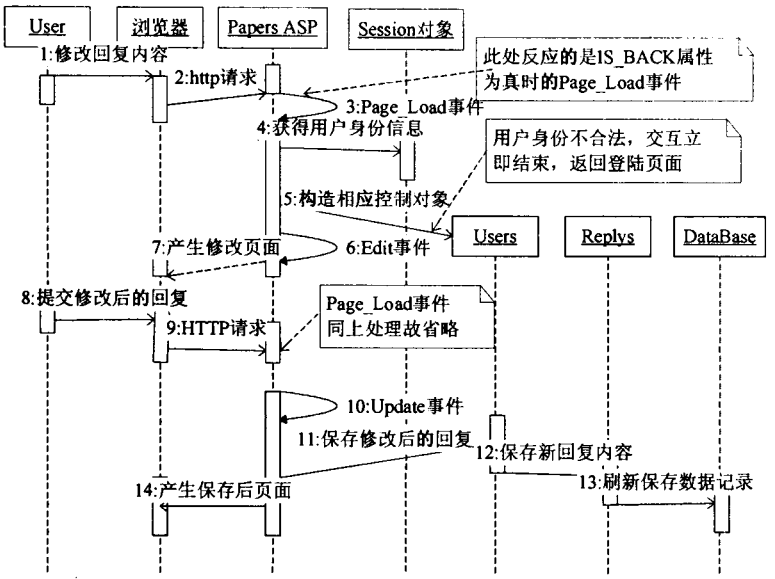

图 1 用户修改自己回复的时序图

1.1 ASP .NET 下 Web 应用表示层的架构

表示层是用户和软件交互的接口，对于 Web 程序设计而言就是基于 HTML 的界面。主要职责就是为用户提供信息，以及把用户的操作传送给逻辑层和数据处理层。从使用案例事件流图中，可以确定用于交互的页面个数，再从案例时序图中，可以确定用于和用户交互的页面和服务器端的 ASPX 页面关系。图 2 是修改自己回复案例用到的页面关系图。

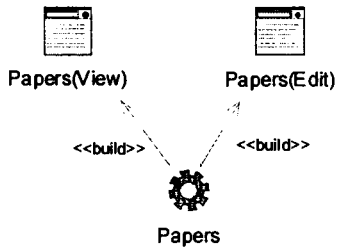

图 2 页面关系图

Papers(View)页面对应用户查看某篇论文的页面，Papers(Edit) 页面对应用户查看自己回复的页面。从案例时序图中可以得知它们都是 Papers 服务器页面在两个不同状态下产生的页面。为了克服传统 Web 窗体页本身无状态这一固有限制，ASP. NET 页框架提供了一种“视图状态”(view state) 的功能，此功能会在往返行程之间自动保留页以及页上所有控件的属性值，ASP. NET 这种特性为数据表现层设计提供了极大的便利。ASP. NET 还采用了由服务器控件引发的事件的工作方式。Web 窗体控件事件模型要求在客户端捕获事件信息，并且通过 HTTP 发送将事件消息传输到服务器，框架再解释该发送以确定所发生的事件，然后在要处理该事件的服务器上调用代码中的适当方法。通过上面的两项技术 ASP. NET 框架提供了可以创建传统客户端/服务器 Web 交互的抽象模型，使开发者能够使用支持快速应用程序开发 (RAD) 和面向对象编程 (OOP) 的传统方法和工具来进行应用程序编程。因此，可以根据角色与系统交互的事件流图和页面关系图来架构 ASP .NET下 Web 表示层：

(1)从页面关系图出发，确定系统中用到的主要功能界面。

(2)根据 ASP. NET 中 Web 控件的特点，对确定的功能界面进行可视化页面布局。

(3)从事件流图出发，确定功能页面之间以及页面状态之间转移逻辑关系，根据 ASP. NET 中 Web 页面级状态管理功能和服务器控件属性、事件编程模式，编写部分页面之间以及页面状态之间转移代码。

在. NET 开发工具中，可以使用“所见即所得”页面设计工具对整个页面进行可视化布局，在实现这些页面之间以及页面状态之间转移逻辑关系代码时，可以在 Web 窗体设计器的“设计”视图中，通过修改对象属性或编写事件完成页面上的逻辑关系代码。

1.2 ASP. NET 下 Web 业务逻辑层的架构

业务逻辑层是整个 Web 应用系统信息和逻辑处理中心，在时序图中反应为由负责时序图流程的控制对象构成。业务逻辑层也是联系 Web 应用系统表示层和数据存储访问层的纽带，因此 Web 业务逻辑层在整个系统的架构中至关重要。从案例时序图中，可以确定控制对象以及控制对象与其它对象所需提供的服务。图 3 是“修改自己回复”案例的时序图中的 users 控制对象的类图。

由于 ASP. NET 提供了一种所谓的后台代码，可用于分离用户界面和逻辑代码，而 ASP. NET 本身也完全支持基于模块与组件开发，因此，为采用面向对象的技术架构逻辑层提供支持，可以从系统用例图和时序图出发在 ASP. NET 下架构 Web 业务逻辑层：

(1)大多数 Web 应用系统都是属于信息管理系统，所以控制对象可以按照使用系统的角色进行划分控制类。

(2)从案例时序图出发，确定控制对象所需的功能。

(3)采用. NET 组件方式包装 Web 业务逻辑层中的功能，从而使逻辑层和表示层在物理上分开。

在. NET 开发工具中，可以采用“类”文件的方式实现 Web 业务逻辑层中的组件，在工程项目中可以采用添加引用的方式把“类”文件引入 Web 工程中，这样，Web 业务逻辑层的功能就可以以对象的方式在应用 Web 系统的开发过程中使用。

1.3 ASP. NET 下 Web 数据访问层架构

数据访问层作为业务逻辑层访问数据存储层的数据访问接口，其主要职责是为数据存储层进行抽象封装，使数据存储层从业务逻辑层看来能完全透明。从案例时序图可以确定实体对象、实体对象的属性及实体对象为控制对象所需提供的服务。图 4 是“修改自己回复”案例的时序图中“Replys”实体对象的类图。

(下转第 60 页)

点管理机制系统。它按照对等网络中节点的网络特性对节点进行分组,分组内节点按照传输性能进行分层,通过赋予网络自组织、自适应的功能来增强组播系统的传输性能。在一个大的对等网络中,节点根据分组标准自主形成中小规模的组播群,数据传输主要集中在组内进行。组内管理采用动态集中式控制,主节点掌握了组播组内成员的全局信息,节点通过彼此交换信息动态调整组内位置,使得分组内的数据传输性能最优化,从而加强了整个网络的性能。另一方面,由于采取了分组的机制,服务器仅需维护分组信息即可而不用管理所有节点,节点的加入、管理和数据传输等完全由节点间 P2P 完成,极大地减轻了服务器的负担,提高了可服务的节点数目。

采用这种节点分组、组内分层的节点自组织管理机制,能够调整各节点在对等网络中的位置,满足异构网络中各节点所处的不同网络环境。通过组内动态调整,使得输出带宽大的节点处于分组的顶端位置,加快数据的分发从而提高多播的效率以减少数据延时。

采取这种上层以 tree 与下层以 mesh 为主体的混合结构,极大地增强了网络的鲁棒性。当网络条件变化时,整个网络会随之动态调整。如新节点的退出或失效不会导致转发路径的断裂,转发路径具有自我修复、自我最优化的能力。因此,这种节点管理机制可以应用于大型文件的共享、流媒体等多种场合。

在实际的应用中,面对复杂的网络环境,如何较为准确地判断节点间的网络距离,进一步强化分组算法和组内管理算法,还有很多工作要做,这是在未来的工作中重点研究的内容。

参考文献:

[1] Banerjee S, Bhattacharjee B. A Comparative Study of Application Layer Multicast Protocols[Z]. US: University of Maryland, 2002.

[2] Francis F. Yoid: Extending the Multicast Internet Architecture, White paper[EB/OL]. http://www.aciri.org/yoid/.. 1999.

[3] Zhang B, Jamin S, Zhang L. Host multicast: A framework for delivering multicast to end users[A]. Proceedings of the IEEE INFOCOM 2002. Vol. 3[C]. New York: Institute of Electrical and Electronics Engineers Inc, 2002. 1366 - 1375.

[4] Chu Y H, Rao S G, Seshan S, et al. Enabling Conferencing Applications on the Internet using an Overlay Multicast Architecture[A]. Proceedings of SIGCOMM 2001[C]. San Diego, CA: [s.n.], 2001.

[5] Chu Y H, Rao S G, Zhang H. A Case for End System Multicast[A]. Proceedings of ACM SIGMETRICS'00[C]. Santa Clara, CA: [s.n.], 2000. 1 - 12.

(上接第 56 页)

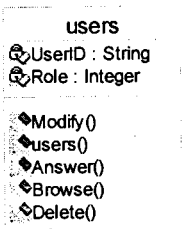

图 3 控制对象的类图

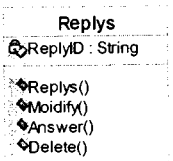

图 4 实体对象的类图

1.4 ASP.NET 下 Web 数据存储层架构

数据存储层的主要功能是把数据访问层的数据处理功能转换为具体的数据库或文件操作。从案例时序图中,可以确定实体对象以及实体类型的属性。实体类中的每个属性可以与数据库中的字段相对应,即每个实体类可以与数据库中的一张表对应。因此,在 ASP.NET 的 Web 应用系统中,可以通过 System.Data.OLEDB 命名空间或 System.Data.SQLClient[4,5] 命名空间中的数据库访问和控制类型构造与具体数据库相适应的类。在数据存储层对实体对象的属性和服务的访问中,可以通过这个类完成对数据库的访问和控制。

2 结 论

采用四层架构的思想,并运用 UML 的对象特点进行类的划分的架构设计,使得代码编写逻辑清晰,易于管理和维护,并且具有很好的代码的可重用性、适用性、易维护性和可移植性。四层架构的编写完全可以由四组人员同时进行,这样代码的维护管理上就更加清晰,并且可以缩短开发周期。但是同时进行的四层代码编写,需要有良好的前期的需求分析的支持。只有完备的需求分析(例如:使程序设计人员都清楚项目所包含或者项目中需要的类名和功能名称,当项目统一使用的时候,就不会因为名称不统一导致引用错误等问题),才可能真正实现四层架构的并行设计。

参考文献:

[1] 飞思科技产品研发中心. ASP.NET 应用开发指南[M]. 北京:电子工业出版社,2002.

[2] Boggs W, Boggs M. UML 与 Rational Rose 2002 从入门到精通[M]. 邱仲潘 等译. 北京:电子工业出版社,2002.

[3] 吴 建,郑 潮,汪 杰. UML 基础与 Rose 建模案例[M]. 北京:人民邮电出版社,2004.

[4] Kauffman J, Matsil B. ASP.NET 数据库入门经典[M]. 张哲峰,黄翔宇译. 北京:清华大学出版社,2003.

[5] Esposito D. 构建 Web 解决方案—应用 ASP.NET 和 ADO.NET[M]. 梁 超 译. 北京:清华大学出版社,2002.