

# 基于 Web 的应用程序构造模式比较研究<sup>\*</sup>

王映辉

(陕西师范大学计算机科学学院 西安710062)

## Compare Research of Web Application Program Organization Pattern

WANG Ying-Hui

(Computer Science College, Shanxi Normal University, Xi'an 710062)

**Abstract** Application development based Web is main-direction at present for application software development, such as enterprise management information system and electronic business software. It is based on client and server. There is three pattern of CGI and special API and Java Servlet for Server application development, and three pattern of Plug-in, Java Applet and ActiveX for client application development. Those pattern's theory, virtue and shortcut are Compared and researched in this paper.

**Keywords** Web application, Organization pattern, Client/server structure, Large scale software

## 1 引言

互联网改变了人们对信息的获取、传输、发布、共享、应用和可视化等的过程和方式,为应用程序数据提供者在 WWW (World Wide Web)上给予了方便的信息发布与共享的方式。目前,几乎所有的管理信息系统都是建立在 Web (或 Internet)技术之上的,它们的构建属于分布式应用开发和大规模软件构架的范畴。其中,客户机和服务器是两个比较重要的概念。在客户机和服务器的配合下,信息的处理不仅是由某一方来完成,而需要多方的合作。由此可知,这种软件的构架完全不同于单机应用程序的构架。

根据任务匹配量的不同,将实现 Web 的应用程序划分为基于服务器端的 (Server-side) 和基于客户机端的 (Client-side) 应用。基于服务器端的应用主要依赖于服务器上的应用系统来完成整个系统的工作,Web 浏览器只充当前端的用户接口,目前的应用系统构造模式主要有 CGI、专用 API 和 Java Servlet 三种。基于客户机端的应用对数据的处理可以在客户机端进行,虽然应用程序最初驻留在服务器上,但当用户向服务器发出请求时,服务器将所需要的数据或数据处理程序传送给客户机端,在客户机端启动数据处理程序并进行相应的数据处理,此时无需服务器的参与,目前的应用系统的构造模式主要有 Plug-in、ActiveX 和 Java Applet 三种。

由于 Web 应用的构建已成为目前企业信息系统等大规模软件构建的主流技术<sup>[1~6]</sup>,而构造模式是这种主流中的核心。因而,对 Web 应用构造模式的比较和研究,有其重要的现实意义和应用价值。

## 2 六种构造模式的工作机理

由上可知,服务器端的 Web 应用系统的构造模式主要有 6 种,下面对这几种常见的构造模式的工作机理进行较详尽的描述。

### 2.1 CGI 模式的工作机理

通用网关接口 CGI 是一组关于如何在 Web 浏览器、Web 服务器与 CGI 应用程序之间传递信息的规范,是超文本标识语言 HTTP (Hyper Text Transfer Protocol) 服务器与程序进行通讯和交互的一种方式<sup>[7]</sup>。在许多应用中,由于 HTTP 不能完全满足用户的需要,如在 WebGIS 的开发中不能直接支持矢量的 GIS 数据,因而需要建立应用服务器。而 Web 浏览器不能直接与应用服务器程序通讯,因此需要有媒介连接 Web 浏览器、Web 服务器和应用服务器。这种媒介中最常见的是 CGI,它接受用户的输入,将其解析为应用程序能使用的变量参数,使得 Web 服务器能在应用服务器上运行,同时解释应用服务器产生的结果,并将结果送回到客户机浏览器。由此可见,应用服务器通过 CGI 与 Web 服务器相连。基于 CGI 的 Internet 应用程序是基于 HTML 的扩展,需要在后台运行应用服务器。应用服务器和 Web 服务器通过 CGI 脚本连接,其体系结构如图 1 所示。客户机端的所有操作,都是在应用服务器上完成的。

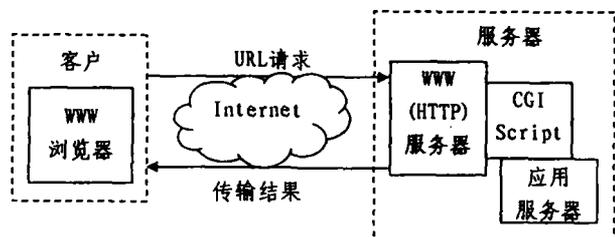


图1 基于 CGI 模式的 Web 应用体系结构

Web 浏览器用户发出 URL 及数据操作请求;Web 服务器接受请求,并通过 CGI 脚本,将用户的请求传送给应用服务器;应用服务器接受请求,进行数据处理;将操作结果通过 CGI 脚本、Web 服务器返回给 Web 浏览器显示。

### 2.2 专用 API 模式的工作机理

CGI 是 Web 服务器调用外部程序的接口,当用户发送一个请求到 Web 服务器,Web 服务器通过 CGI 把该请求转发

<sup>\*</sup> 国家自然科学基金(60072044)和陕西省自然科学基金资助项目(2000X14)。王映辉 博士,副教授,目前的主要研究方向为可视化技术与大规模软件工程。

给后端运行的应用服务程序,由应用服务程序将生成的结果交给 Web 服务器,Web 服务器再把结果传递到用户机端显示。这种方法的缺点是,对于每一个客户机的请求都要重新启动一个新的服务进程,其效率受到一定的影响。

针对 CGI 程序的上述不足之处,为了提高 Web 服务器与应用服务器之间的通信效率和性能,各大 Web 服务器厂商和专用应用厂商(如数据库厂商)纷纷推出各自的专用 API。在 Web 服务器与应用服务器的连接方案中,Netscape 和微软作为 Web 服务器厂商分别推出了适用于各自 Web 服务器的 NSAPI(Netscape Server Application Programming Interface)和 ISAPI(Internet Server Application Programming Interface)。

专用 API 模式可以实现 CGI 的全部功能,并对之进行了扩展,其工作机理与 CGI 大致相同,都是通过交互式网页取得客户输入信息,然后交服务器后台处理。

由于专用的 API 一般采用动态连接库 DLL 的形式,而动态连接库可以和服务器装于同一地址空间中,因此执行效率比 CGI 高。

对于 ISAPI 来说,提供了两种简单有效的方法来扩展 Web 服务器:一种称作 ISAPI 服务器扩展,另一种称为 ISAPI 过滤器<sup>[7]</sup>。ISAPI 服务器扩展是一个动态连接库,可以被 Web 服务器作为一个客户程序调用和加载;ISAPI 过滤器类似于 Internet 信息服务器 IIS,用于过滤传给服务器或从服务器传出的数据。

ISAPI 过滤器辅助处理 http 请求过程中的各种事件,注册事件、通知、消息。当选择的事件发生时,过滤器被调用并可以监视和更改数据。利用 ISAPI 过滤器可以使 Web 服务器增加各种不同的功能,如验证、压缩或加密等。

### 2.3 Java Servlet 模式的工作机理

Java Servlet(简称 Servlet)是在服务器端运行并可扩展服务器功能的 Java 程序。Servlet 使用 Java Servlet 的 API 及其类和方法来完成与用户的交互。当 Servlet 接收来自用户的请求时,通过两个对象 ServletRequest 和 ServletResponse 来响应和完成服务器端与客户机端的通讯。ServletRequest 接口可以提供客户机端传来的参数名、客户机端使用的协议、产生请求的主机名和接收请求的服务器名等。ServletResponse 所提供的方法能够完成对客户机端的响应,包括设置内容的长度和应答类型等。

当客户机端有请求时,服务器加载并初始化 Servlet,接收来自客户端的请求,响应请求,提供服务,并将结果返回给客户机端,最后卸载 Servlet。具体来说,当客户请求到达服务器后,服务器创建两个对象:请求对象 ServletRequest 和响应对象 ServletResponse;之后,服务器调用 Servlet 的 service()方法来从请求对象处获得请求信息;处理该请求并通过响应对象将结果传回客户机端。一个 Servlet 能同时运行多个 service()方法,但每个 service()方法只对应一个客户机请求。

使用 Java Servlet API 可以开发服务器端的组件,为服务器增加一些强大的功能。另外,这种开发往往与基于客户端的 Applet 配套使用,共同完成相应的功能。

### 2.4 Plug-in 模式的工作机理

Plug-in 是在浏览器上扩充 Web 浏览器的可执行的软件,它必须安装在客户机,然后才能使用。Plug-in 直接处理来自服务器(如数据库服务器)的应用数据,同时可以生成自己的数据,以供 Web 浏览器或其它 Plug-in 显示使用。由此可

见,Plug-in 的主要作用是使 Web 浏览器支持特定应用程序的功能,是 Web 浏览器与特定应用程序之间通讯的桥梁和纽带,其体系结构如图2所示。

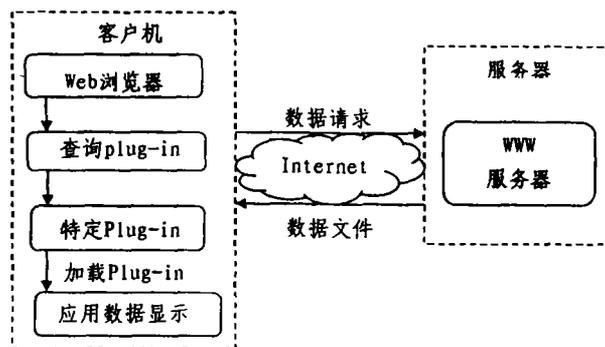


图2 基于 Plug-in 模式的 Web 应用体系结构

首先,Web 浏览器发出应用数据显示操作请求;当 Web 浏览器受到用户的请求,进行处理,并将用户所要的数据传送给 Web 浏览器;客户机端接受 Web 服务器传来的数据,并将此数据类型进行理解;之后,在本地系统查找与应用数据相关的 Plug-in。如果找到相应的 Plug-in,用它来处理 and 显示数据;如果没有,则需要安装(一般情况下自提供数据的服务器上可以下载并安装)相应的 Plug-in,并启动,进行相应的处理数据。由此可见,在客户机端可以由 Plug-in 来完成对数据的加工和处理任务。

### 2.5 ActiveX 模式的工作机理

ActiveX 是微软公司为适应互联网而发展的标准,是建立在动态连接与嵌入 OLE(Object Linking and Embedding)标准之上,为扩展微软 Web 浏览器 Internet Explorer 功能而提供的公共框架。ActiveX 控件是用于完成具体任务和信息通讯的软件模块。采用 ActiveX 控件对其功能进行扩展是目前在 Windows 操作系统之上开发软件普遍采用的技术,也是 Windows 操作系统之软件模块得以复用的核心规范<sup>[10,11]</sup>。

基于 ActiveX 控件的互联网应用系统是依赖 ActiveX 来完成对应用数据的处理的。ActiveX 控件与 Web 浏览器无缝结合在一起。在通常情况下,ActiveX 控件包含在 HTML 代码中,并通过<OBJECT> 参考标签来获取。

当 Web 浏览器发出应用数据显示操作请求时,Web 服务器接受到用户的请求,进行处理,并将用户所要的数据或 ActiveX 控件(如需要的话)传送给 Web 浏览器;之后,客户机端接受到 Web 服务器传来的数据和 ActiveX 控件,启动 ActiveX 控件对相应数据进行处理,并完成各种操作(图3)。

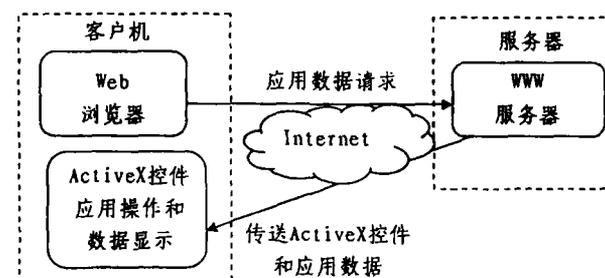


图3 基于 ActiveX 控件的 Web 应用体系结构

ActiveX 控件和 Plug-in 非常相似,都是为了扩展 Web 浏览器的动态模块。二者不同的是,ActiveX 能被支持 OLE 标准的任何程序语言或应用系统所使用,而 Plug-in 只能在

某一具体的浏览器中使用。

## 2.6 Java Applet 模式的工作机理

在通常情况下,Java Applet 包容在 HTML 代码中,并通过<APPLET>参考标签来获取和引发<sup>[8,9]</sup>,能完成应用数据解释处理功能。但是,对于处理大量的任务,无法与 CGI 模式相比。Java Applet 是由面向对象语言 Java 开发的小应用程序,与 Web 浏览器紧密结合,以扩展 Web 浏览器的功能,完成应用数据操作和处理。

Java Applet 最初为驻留在 Web 服务器端的可执行代码,它在程序运行时,从服务器下载到客户端并加以运行。其工作机理图与 ActiveX 控件相似。

## 3 6种构造模式的优缺点

### 3.1 优点

对于 CGI 模式,首先它是一种“胖”服务器、“瘦”客户端的模式,有利于充分发挥和利用服务器的最大潜力;另外,由于在客户端使用的是 Web 浏览器,因而客户端与平台无关。对于专用 API 模式,首先由于所有的应用操作和分析都是在应用服务器上完成的,客户端任务匹配很小,因而也是一种“胖”服务器、“瘦”客户端模式,可以充分发挥服务器的资源和强大的处理功能;另外,由于在客户端使用的是支持标准 HTML 的 Web 浏览器,因而客户端与平台无关;此外,由于专用 API 大多采用动态连接库 DLL 的形式,可以和服务器装于同一地址空间中,因此执行效率比 CGI 高。

利用 Servlet 模式进行的开发,首先是可以充分发挥服务器的资源和强大的处理功能;另外,由于每个请求生成的线程都在一个进程中得到了服务,因此响应速度比 CGI 高;此外,在开发中大多采用 Java 语言作为工具,因此继承了 Java 语言的优点,如程序功能的组件化、与软件版本和运行环境(软件与硬件)的无关性、高安全性等;再者,Java Servlet 与 Java Applet 的配合使用,是目前基于 Internet 的开放分布式负载均衡软件开发的理想模式。

对于 Plug-in 模式,首先 Plug-in 与每一种数据源是一种无缝的连接,即它们之间紧密相关;其次,所有对数据的操作都是在客户端本地由 Plug-in 完成,因此运行的速度快;再次,当服务器仅需提供应用数据服务时,网络也只需将数据一次性传输,服务器的任务量减少,网络传输的负担轻。

对于 ActiveX 模式,除了具有 Plug-in 模式的所有优点之外,还能被支持 OLE 标准的任何程序语言或应用系统所使用,比 Plug-in 模式更灵活,使用方便。

对于 Java Applet 模式,首先由于 Java Applet 在具有 Java 虚拟机的 Web 浏览器上运行,因而具有与软硬件无关性;其次,由于 Java Applet 是在运行时从 Web 服务器动态下载的,无须在用户机端预先安装,即可及时运行;再次,当服务器端的 Java Applet 更新后,客户端总是可以使用最新的版本;另外,由于所有的操作都是在本地由 Java Applet 完成,因此运行的速度快;此外,Applet、Java 语言与公共代理框架模型 CORBA 的结合是目前移动 Agent 软件开发的理想工具。

### 3.2 缺点

对于 CGI 模式,首先由于用户的每一步操作都需要将请求通过网络传给应用服务器,应用服务器将操作结果形成新的信息格式,再通过网络返回给用户,因而大大增加网络传输的数据量,所以网络传输的负担重;其次是服务器的负担重;再次,由于客户端的功能受 Web 浏览器的限制,因此用户

界面的灵活性差;此外,由于 CGI 模式对于每一个客户机的请求都要重新启动一个新的服务进程,当有多个用户同时发出请求时,存在并发多请求问题,系统的性能和工作效率将受到影响;同时,客户端的功能还受到互联网浏览器功能的限制。

对于专用 API 模式,首先是网络传输和服务器的负担重;再次是用户界面的灵活性差,客户端的功能还受到 Web 浏览器功能的限制;另外,由于某一专用的 API 与其它专用 API 之间互不兼容,它们只能在特定的服务器和操作系统上运行,因此移植性差;同时由于采用了动态链接库的形式,一旦代码质量较差,就比较容易造成服务器系统的崩溃,因此在进行程序设计时会更复杂。

在利用 Servlet 模式来开发应用程序时,由于安全因素,对数据的获得、保存和分析以及结果的存储等,紧紧依赖于网络,即受到网络资源的能力的限制;另外,Servlet 的运行需 Java 虚拟机的支持,其运行速度受到一定的影响;此外,容易出现服务器负载过重的现象。

对于 Plug-in 模式,首先对同一应用数据,对于不同的操作系统、不同的 Web 浏览器和不同的用户数据,需要有各自不同的 Plug-in,因而 Plug-in 与平台和数据相关;其次,需要事先下载和安装 Plug-in 程序并预先安装并占有客户机额外的磁盘和内存空间;此外,在升级时,由于必须通知每一个用户重新下载和安装,因此更新比较困难。

对于 ActiveX 模式,首先是需要下载,占用客户端机器的磁盘和内存空间;其次是与平台和浏览器相关,对于非 Microsoft Web 浏览器,必须增加特殊的 Plug-in 来支持相应的 ActiveX 的运行。

对于 Java Applet 模式,首先是完成大型应用分析和处理的能力有限;此外,由于安全因素,Java Applet 对数据的获得、保存和分析结果的存储等,紧紧依赖于网络资源的能力,即它的运行受到网络的限制;最后要说明的是,Java Applet 与 Java Servlet 一样,在运行时都需 Java 虚拟机的支持,其运行速度受到一定的影响。

**结束语** 对于 Web 应用来说,与传统的应用的最大的不同是客户端的程序主要在常用的浏览器中运行,如 IE、Netscape 等,扩展了应用的兼容性,大大促进了应用程序走向开放。在基于服务器的构造模式中,CGI 是一种较早的构造模式,而专用 API 和 Java Servlet 包括了 CGI 的全部功能,运行效率也比较高。

ActiveX 控件、Java Applet 在页面中的使用扩展了客户端浏览器的功能,可以实现浏览器无法实现的功能,许多任务不必再求助服务器,从而大大减小了网络和服务器器的负担。ActiveX、Java Applet 均是以构件的形式出现的,可以灵活配置,而且可以回调服务器上其它构件的方法,在很大程度上增强了应用的兼容性和开放性。

ActiveX 控件目前只有 IE 全面支持,要使用在非 Microsoft 公司的 Web 浏览器中,则必须有特制的 plug-in 的支持才能运行,兼容性较差;Java Applet 以其平台独立性、安全性好等特性,在 Web 应用开发中得到了广泛的应用,但是 Java 运行的速度较慢,每次都需下载,比较费时,对客户端的要求较高。

特别值得一提的是,在 Java 语言的支持下,Java Servlet、Java Applet 和 CORBA 的融合,是目前基于移动 Agent 软件

(下转第 138 页)

并把 B 的地址传给 A。A 在获得教师 B 的地址后,可直接向 B 发出学习请求,B 给予响应并开始学习,学习过程中不再需要其它 Facilitator 介入。

## 5 系统的实现

考虑到远程教学系统的分布特性和开放性,我们采用方便维护升级的基于 Web 的浏览器/服务器模式,学生客户端采用 IE 或 Netscape 标准浏览器,服务器端采用 WINDOWS 2000 IIS 5.0 Web 服务器,信息发布系统采用 VBScript 和 ASP 脚本网页数据库交互式应用技术开发,部分网页采用 JavaScript 技术实现,后台数据库采用 SQL Server 2000 管理各种数据表格,智能导航算法和搜索教师 Agent 以及学习效果自我评价、考试测试系统都是采用 ASP 和 ActiveX 技术开发的。访问权限管理除了一般的用户名密码验证外,还增加了 IP 地址访问权限管理以及 S-HTTP 编码传输技术。开发工具采用 Microsoft InterDev 和 FrontPage。在个性化跟踪方面,我们采用基于 MS Agent 的角色编程技术,并与 ActiveX 控件相结合,给予学生以各种人性化的提示和引导。超级管理员、教师和学生分别进入不同的权限画面,并可进行教学资源管理和讨论交流等细化分工。具体实现界面如图 3 所示。

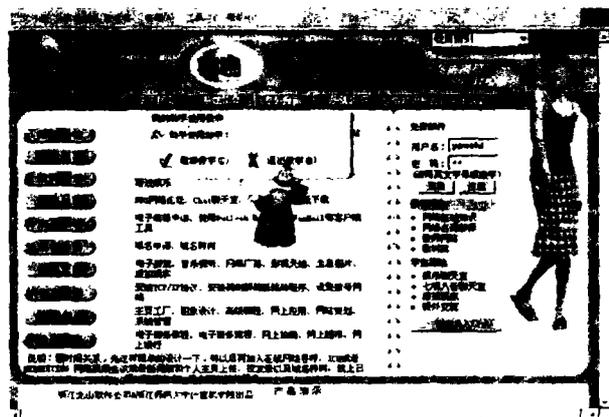


图3 采用 Agent 技术的 Internet 模拟远程教学系统

## 6 国外远程教学 Agent 成功实例

目前,世界范围内正在广泛开展教学 Agent 应用系统的研制,也建立了一些实用的系统,其中有些系统的行为近似智能辅导系统,但还没有真正体现 Agent 思想。比较著名的远程教学系统是美国信息科学研究所高级技术研究中心开发的

ADL (远程教学 Agent, Agent for Distance Learning) 和 STEVE (虚拟环境滑行训练专家, Soar Training Expert for Virtual Environment)。STEVE 用于美国海军舰艇发动机的操作培训中,可在网络环境中与学生进行交互,学生在这里看到的 Agent 是一个 3D 角色,学生可听到它讲话,Agent 可凭借虚拟环境中的跟踪硬件监视学生所处的位置及要进行的操作,其中很多核心技术都是由多学科多专业跨公司协调完成的。STEVE 是个较高层次的应用,而 ADL 更适合国内的应用技术水平,如在对医生的继续教育课程中,基于 Web 的 ADL 会出现在屏幕上进行提问和机会式指导,并回答学生的提问而且具有情感表态,减轻学生的枯燥感。如在学生正确解题后,ADL 面部会出现满意神色等,错误时则会表现焦虑神色。这类角色类编程现在也已经推出了商用开发工具,如微软的 MS Agent 组件开发套件和基于 JAVA 的 Agent Builder 等,开发这类远程教学系统在技术上已经有一定经验储备。

**结论** 基于 Agent 思想的远程教学系统尚处于研究与实验开发阶段。但随着人工智能和信息技术的发展,网络化、智能化和个性化的计算机远程教学模式将成为未来教育的主要手段,其应用前景非常广阔。尤其在现在流行的 Web 技术基础上嵌入人工智能和 Agent 技术,在教学效果、教学模式和系统性能上必将大大优于传统教学系统,对整个教学改革和素质教育的实施起到积极的推动作用。当然,这个领域的研究还有待于进一步深化。相信不久的将来,更多符合中国国情、符合未来学习模式的人性化远程教学系统必将开发成功!

### 参考文献

- 1 蔡斌,周明天,杨国纬. Agent 模型. 计算机应用,1999,19(6)
- 2 赵建民,朱信忠. 基于 Agent 的 Visual C++ COM 编程实现技术. 微电子学与计算机,2002(11)
- 3 于长云,王哲明. 教育代理—远程教学系统课件功能智能化. 天津纺织工学院学报,29(4)
- 4 潘瑞玲,余轮. Agent 技术在远程教育系统中应用的研究. 微型电脑应用研究与设计,2002,18(4)
- 5 曲朝阳,王丽英. 基于 Agent 的远程教育系统研究. 东北电力学院学报,2001,21(2)
- 6 Lew J W. Pedagogical Agents [EB/OL]. <http://www.isi.edu>. 1999. 10. 4
- 7 Finin T, Fritzon R. A Language and Protocol to support Intelligent Agent Interoperability, April 2001. <http://www.cs.umbc.edu/kqml/papers>
- 8 谭淑英,李赫男,左贵启. 服务器端的动态网站开发技术. 计算机应用研究,2002,5:143~149
- 9 王映辉,周明全,耿国华. 基于 WebGIS 的“数字城市”分布式协作模型研究. 计算机工程与应用,2002. 5(38):20~22
- 10 SmithRG, DavisR. Frameworks for cooperation in distributed problem solving [J]. IEEE Transaction on Systems, Man and Cybernetics, 1981, 11(1):61~70
- 11 黄浩文,黄静舒. JSP 核心技术和电子商务应用研究[M]. 北京:机械工业出版社,2001. 3
- 12 木森林,等. Visual C++ 6.0 使用与开发[M]. 北京:清华大学出版社,1998
- 13 陈章渊等译. 智能 CORBA [M]. 北京:电子工业出版社,1999
- 14 Sun Microsystems. Java Servlet Specification Version 2.3 [R]. Sun Microsystems, 2000
- 15 潘爱民,等. COM 原理与应用[M]. 北京:清华大学出版社,2000
- 16 钱力棚,等. Visual InterDev 6.0 网络编程技术[M]. 北京:人民邮电出版社,2000. 1
- 1 王磊,谢俊元. 利用 EJB 构造电子商务平台[J]. 计算机应用研究,2002,5:146~149
- 2 孙艳春. CSCW 系统体系结构中协作机制的研究[J]. 小型微型计算机系统,2001,10(22):1182~1185

### 参考文献

(上接第 116 页)  
开发和实践的理想模式。

总之,本文对基于 Web 的应用程序的几种构造模式的工作机理进行了较详尽的描述,以便我们在基于 Web 的大规模软件开发中,特别是在开放分布环境下的协同智能 Agent 软件开发中,以及基于 Web 应用的多机负载均衡等方面,能将它们融合使用,使开发出的 Web 应用程序在时间和空间上达到最佳。