

IntraWeb 与 ActiveForm 混合编程的实现

Realization of Mixed Programming with IntraWeb and ActiveForm

祝建中 (杭州师范学院信息工程学院 310036)

郑瑜钊 (杭州师范学院图书馆 310036)

摘要:针对 B/S 结构下如何合理规划客户端应用程序、充分利用客户机资源以满足系统需求、提高系统效率的问题,分析讨论了 IntraWeb 和 ActiveForm 混合编程的关键技术。

关键词:IntraWeb ActiveForm 混合编程

1 问题的提出

基于网络和分布式技术的 B/S (Browser/Server) 结构多层分布式系统,以其在性能、管理、扩展等方面的明显优势成为当今最流行的应用系统结构之一。如图 1 所示,系统的数据逻辑由数据库服务器实现,系统应用逻辑由中间层的应用服务器实现,而系统的表示逻辑由 Web 服务器与客户端浏览器共同承担。在基于 B/S 结构的多层体系中,系统表示逻辑的工作机理为:用户通过浏览器中运行的界面向 Web 服务器发出请求(request),Web 服务器中相应的 Web 服务器应用程序扩展(Web server extension,如 CGI、ISAPI 等)接收用户的请求并进行分析处理,然后把处理结果以适合浏览器显示的数据形式响应(response)用户。在这种工作模式下的客户端应用程序虽然具有易维护、易扩展等优点,但因负责实现表示逻辑主要工作的 Web 服务器应用程序扩展是运行在 Web 服务器上,故不能直接使用系统中众多客户机上丰富的软硬件资源,使得系统客户访问量增大时 Web 服务器负载过重,而运行浏览器的客户机只承担 request 的发送和 response 显示。

例如,“学生自主学习系统”主要目标是充分发挥每个学生的特长,使他们能够在不同的时间、地点根据自身当前的学习状态选择学习内容和把握学习进度,基于 B/S 的多层分布式体系结构是开发该系统的合理选择。在这样的系统中,众多学生通过各自的客户机

与系统交互,在学习过程中,客户机除了利用浏览器提供输入/显示界面外,还可能需要使用本地的磁盘文件(如在模拟练习过程中暂存各练习题的答题结果,待积累一定的数量后一并提交以降低网络的通信开销和服务器的响应次数)、打印机(如以可控制的格式打印学习资料)、甚至启动本地的应用程序(如启动某个编程环境以完成某编程题练习)。本文讨论如何通过 IntraWeb 和 ActiveForm 混合编程来解决这类系统需求的技术和方法。

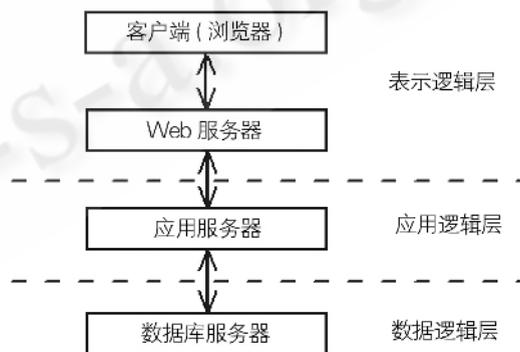


图 1 基于 B/S 的多层体系结构

2 解决方案

2.1 基本思想

Delphi 为基于 Web 的可视化编程提供了强有力的环境和工具,IntraWeb 和 ActiveForm 是 Delphi 编程

环境下两种功能强大、各具特色的 Web 应用开发工具。

IntraWeb 是 AtoZedSoftware 用 Delphi 实现的基于组件的 Web 应用开发构架,可用于 Delphi、Kylix、C++ Builder 和 Java 环境下快速、方便地开发交互式 Web 应用^[1-3]。IntraWeb 的主要长处是:完全支持所见所得的开发方式,具有与开发 Win32 模式下应用程序几乎同等的设计能力。IntraWeb 应用程序有两种模式: Stand Alone Application 和 Web server extension (ISAPI DLL 或 Apache module,见图 2)。IntraWeb 集成了一个小巧的 http Server,使开发人员可以在 Delphi 编程环境中便捷、稳定地调试 Stand Alone Application,一旦调试成功就可方便地将其转换为 Web server extension 并部署到 Web 服务器中。但 Web server extension 以 ISAPI 方式运行在 Web 服务器应用程序的进程空间内,故较难充分利用客户机的资源。

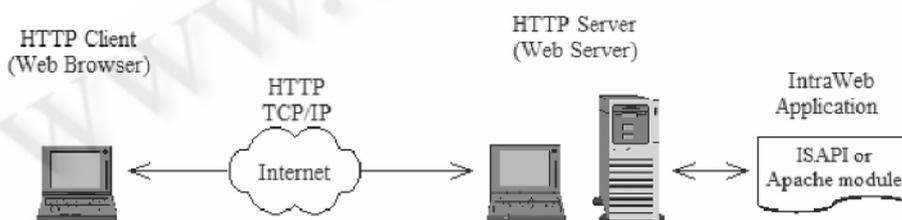


图 2 Web server extension 模式的 IntraWeb 应用程序

ActiveForm 是封装了 Delphi Form 的 ActiveX 组件,可以在任何支持 ActiveX 的应用程序中使用,包括支持 ActiveX 的浏览器。当浏览器运行某个 ActiveForm 时,首先检查是否已经下载到本地,若是第一次运行或该 ActiveForm 的版本已经更新,则先从服务器下载,否则就直接运行。可见,ActiveForm 是下载后驻留在客户机上并在客户机器上运行的,因此 ActiveForm 能够像任何本地进程一样方便地访问客户机上的各类资源。但纯粹采用 ActiveForm 技术开发的客户端应用程序可能存在安全隐患等缺点。

若能根据具体应用的特征,从功能和结构上对客户应用程序进行合理的划分,让其与应用服务器密切相关的部分采用 IntraWeb 技术实现运行在 Web 服务器端,而与客户端资源访问密切相关的部分采用 Ac-

tiveForm 技术实现运行在客户机。两种技术混合应用以达到充分发挥各自的长处、克服各自的不足的目的。

2.2 ActiveForm 嵌入 IntraWeb 的技术

两种技术混合编程的第一个要解决的问题是解决如何将 ActiveForm 嵌入到 IntraWeb 中使其能够访问客户机资源。

首先,根据客户端应用程序的功能划分,新建一个 Stand Alone Application 类型的 IntraWeb 项目(不妨取名为 IWX.dpr),在该 IntraWeb Form 中进行设计客户界面、实现数据表示逻辑中适合在 Web 服务器端运行的各部分的功能。在设计客户界面时,预留以后集成 ActiveForm 所需要的布局空间。

接着,创建一个 ActiveX Library 项目(不妨取名为 IWActiveForm.dpr),在该项目中加入一个 ActiveForm 组件,在该 ActiveForm 组件中设计实现客户端应用程序中需要访问客户机资源的界面及相应的功能。编译

该项目生成 IWActiveForm.ocx 组件后,打开 Project | Web Deployment Options 对话框,在 Target dir、Target URL 和 HTML dir 编辑框中填写与发布相关的路径值,再用 Project | Web Deploy 发布此 ActiveForm(通常的发布位置为嵌入此 ActiveForm 的 IntraWeb 应用程序所在的文

件夹,而 IWActiveForm.ocx 最后将复制到该文件夹,故上述 Target dir、Target URL 和 HTML dir 的值一般应依次设置为“.\”、“./”和“.\”)。发布过程将生成文件 IWActiveForm.html,该文件含有两项与 ActiveForm 嵌入 IntraWeb 相关的重要信息:

```
classid = "clsid:25027BEF - E288 - 43A1 - 9B76 - A2C6492E2E20"
```

```
codebase = ". /IWActiveForm.ocx#version = 1,0,0,0"
```

最后,回到 IntraWeb Application Form 设计时预留的位置,添加一个 IWActiveX 组件并调整其大小与要嵌入的 ActiveForm 相符,将该 IWActiveX 组件的 ClassID 和 CodeBase 属性设置为上述提及的 IWActiveForm.ht-

ml 中 classid 和 codebase 的相应值,经编译、调试通过后,转换成 Web server extension。再把 IWebForm.ocx 复制到嵌入它的 IntraWeb 应用程序所在的目录,并把该目录设置成 Web 服务器的虚拟目录即完成发布。当该 IntraWeb 应用程序在 Web 服务器上运行时,嵌入的 ActiveForm 将被下载安装到客户机并在客户机上运行,这样就实现了系统表示逻辑的一部分在 Web 服务器上运行,而另一部分在客户机上运行的目的。

2.3 IntraWeb 与 ActiveForm 通信技术

IntraWeb 与 ActiveForm 混合编程的另一个关键问题是如何实现两者的通信。因为 IntraWeb 运行在 Web 服务器上而 ActiveForm 运行在客户机,所以可以利用远程方法调用机制来实现两者间信息传递。

以“学生自主学习系统”为例,在用 IntraWeb 技术设计的 Web 页面 IWebForm1 中,组件 Table1 关联的数据表记录具有“学习主题”、“参考文献”等字段,为了便于学员在需要时能够在客户端打印机上打印当前学习主题(设显示在 IWebMemo1 组件中)及其相关的参考文献目录,在 ActiveForm 设计时设置一个 TRichEdit 组件 RichEdit1 用于暂存和打印这些信息。为此,在设计 ActiveForm 时,必须利用 View I Type Library Editor 扩展 ActiveForm 的对外接口 IWebFormX,即在 IWebFormX 中增添一个可供嵌入该 ActiveForm 的 IntraWeb 应用程序调用的远程方法 AddToRichEdit,并加以实现:

```
procedure TIWebFormX. AddToRichEdit ( const
aMsg: WideString );
begin
RichEdit1. Text := aMsg;
end;
```

另一方面,在用 IntraWeb 技术设计的 Web 页面 IWebForm1 中,编写在浏览器内执行的、调用 AddToRichEdit() 方法的 JavaScript 代码。例如,我们希望双击显示在 IWebMemo1 组件中的“学习主题”时,就能够将该“学习主题”及其相关的“参考文献”复制到客户端的 RichEdit1. Text 中,则可按如下所示编写 IntraWeb 的 Web 页面 IWebForm1 的 OnRender 事件处理代码:

```
procedure TIWebForm1. IWebFormRender ( Sender:
```

```
TObject );
begin
IWebMemo1. ScriptEvents. Clear;
with IWebMemo1. ScriptEvents. Add ( 'onDbClick
') do
begin
EventCode. Clear;
EventCode. Add ( 'IWACTIVEX1. Ad-
dToRichEdit(" ' +
'学习主题:' + Table1. FieldByName ( '学习
主题') . AsString + '\n' +
'参考文献:' + Table1. FieldByName ( '参考
文献') . AsString + " " ); )
end
end;
```

需要注意 WACTIVEX1 是大小写敏感的,而方法名 AddToRichEdit 的大小写是无所谓的。信息一旦复制到客户端,就可方便地调用 RichEdit1. Print 方法进行本地打印了。

3 结束语

实践表明,利用 IntraWeb 和 ActiveForm 混合编程技术,能够在几乎没有削弱 B/S 模式原有优点的前提下,较好地解决 B/S 应用模式下客户端本地打印、临时目录及文件的建立等问题,有利于充分利用系统资源,提高系统的整体效率。

参考文献

- 1 Cary Jensen. IntraWeb: A New Way to the Web. <http://bdn.borland.com/article/0,1410,29650,00.html>. 2003.
- 2 ATOZED. Why IntraWeb? <http://www.atozed.com/intraWeb/Features/index.iwp>.
- 3 ATOZED. Two Kinds of Web Development. <http://www.atozed.com/intraWeb/Weblication/index.iwp>.