

一种基于 WebDAV 协议的 OFFICE 文档 在线编辑实现方法^①

An Implementation Method for WebDAV Protocol-Based Office Files to Be Edited Online

段江涛 白 莉 赵 俊 (云南省科技情报研究院 650051)

摘 要: WebDAV (Web-based Distributed Authoring and Versioning) 是基于 HTTP 1.1 的一个通信协议, 它为 HTTP 1.1 添加了一些扩展, 使得应用程序可以直接将文件写到 Web Server 上。基于 WebDAV 协议提出了一个 OFFICE 文档在线编辑的实现方案, 可实现对服务器端的 OFFICE 文档进行在线编辑, 同时讨论了实现过程中与 WebDAV 协议相关的技术细节。

关键词: WebDAV OFFICE 文档 在线编辑 .net

1 概述

随着办公自动化技术的发展, 协同工作成为办公自动化技术中一项较为关键和重要的技术。计算机支持的协同工作(CSCW)是指在计算机技术支持的环境下, 一个群体协同工作完成一项共同的任务(CW), 其目标是涉及支持各种各样的协同工作的应用系统。文档的在线编辑属于协同工作中协同写作和讨论系统(Co-authoring and argumentation System)的范畴^[1]。其主要技术包括: 文档的在线编辑、版本控制、痕迹跟踪、文件的锁定等。

WebDAV(Web-based Distributed Authoring and Versioning)是基于 HTTP 1.1 的一个通信协议, 其技术规范由 IETF RFC 2518 定义。它为 HTTP 1.1 添加了一些扩展, 使得应用程序可以直接将文件写到 Web Server 上。由于 WebDAV 协议使用了 XML 技术, 并可以利用 HTTP 的诸多成熟的特性, WebDAV 协议可以在一些应用上替代 FTP 协议。微软多数产品均实现了对 WebDAV 协议良好的支持, 以下讨论如何使用 WebDAV 协议支持实现对 OFFICE 文档的在线编辑。

2 基于WEBDAV协议的OFFICE文档在线读写实现原理

WebDAV 本身是一个类似于 HTTP 协议的通信协议(IETF RFC 2518)。它与 HTTP 协议类似, 需要服务器和客户端两个部分的实现。本文中, 客户端部分使用 OFFICE(至少 2003 以上版本), 服务器端使用 IIS(至少 6.0 以上), 在 .net Framework 下实现。

2.1 客户端实现原理

OFFICE 从 2000(包括 OFFICE XP)以上的版本即对 WebDAV 协议有良好支持。自 OFFICE 2003 以后的版本, 此功能又得到了很大的加强。在安装 OFFICE 2003 或更新的版本后, 就会自动安装一个 ActiveX 控件到系统中, 这个控件在 `owsclt.dll` 中定义^[2]。使用脚本调用此 ActiveX 控件即可激活 OFFICE 2003/2007 打开在 Web 上的 OFFICE 文档, 在打开文档及修改编辑的全过程中, 此 ActiveX 控件将在 WebDAV 协议下与 OFFICE 2003/2007 一同负责与服务器的通信。例如, 打开并编辑一个在服务器端的文档可以使用如下脚本:

^① 收稿时间:2009-03-10

```
myDoc=new ActiveXObject("SharePoint.
OpenDocuments.2");
myDoc.EditDocument(url,strFileType);
```

参数 SharePoint.OpenDocuments.2 表示使用 OFFICE 2003 作为客户端打开文档，如使用 OFFICE 2007，则为 SharePoint.OpenDocuments.3。参数 url 表示文档的地址；strFileType 为 OFFICE 文档的类型，为预定义的字符串，word 文档即为 Word.Document。

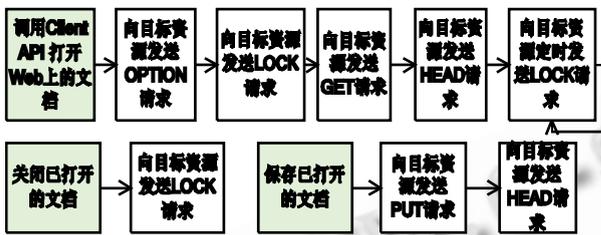


图 1 客户端与服务器端交互过程

使用抓包工具拦截并分析此脚本调用 ActiveX 激活 OFFICE 客户端打开位于 Web 上的文档至编辑状态的过程，发现客户端与服务器端的交互过程如上图 1 所示。打开 WEB 上的文档即顺序执行上图所示的请求。OPTIONS、GET 及 HEAD 请求是标准的 HTTP 1.1 请求，LOCK 请求是 WebDAV 协议特有的请求。其中，OPTIONS 请求返回的是服务器支持的请求类型；GET 请求返回的是资源实体，即待编辑的 OFFICE 文档；HEAD 请求返回的是资源信息，例如文档类型，资料长度等；LOCK 请求则返回资源已经被成功锁定的消息。LOCK、GET 及 HEAD 请求是必须的，服务器端必须返回响应，否则不进行下一步请求，文档无法打开进行编辑。完成 HEAD 请求后，文档打开，进入正常编辑状态。此后每隔 2 分 15 秒，OFFICE 客户端将会自动向服务器端发送 LOCK 请求，如果不能得到正常响应，则编辑终止，客户端将显示无法打开文档。

在保存已打开的 OFFICE 文档时，OFFICE 将发送一个 PUT 请求，并将正在编辑的文档放在消息体中发送到服务器。客户端收到成功保存的响应(200 OK)后，再发送一个 HEAD 请求，返回保存后文档大小等信息。

在关闭已打开的 OFFICE 文档时，关闭之前 OFFICE 将发送一个 UNLOCK 请求，解除对文档的锁定，允许其它用户对文档进行编辑。此请求也是 WebDAV 协议特有的请求。

值得注意的是，如果不是调用 ActiveX 控件激活 OFFICE 2003/2007 客户端来打开 Web 上的文档，例如直接用 OFFICE 2003/2007 打开一个位于 Web 上的资源(前提是该服务器支持 WebDAV 协议)，则 OFFICE 客户端不会执行上述请求。

2.2 服务器端实现原理

通过以上对客户端实现原理的分析，可以得知服务器端需要完成的工作有两项：一是在收到客户端的请求后，对请求内容进行操作；二是在操作内容完成后，给客户端以正确的响应。只需实现如上所述的常用的几个响应操作，即可对 OFFICE 文档进行在线编辑。下面以表格的形式给出服务器在接收到请求后完成的操作及做出的响应。

表 1 服务器端操作及响应

客户端请求 (Request)	服务器端操作 (Operation)	服务器端常见响应 (Responses) (错误响应代码与 HTTP 同, 不再列出)
OPTIONS	将服务器端预定义支持的请求发送给用户。	200(OK), 返回服务器支持的 WebDAV 协议请求列表;
GET	打开 OFFICE 文档并传送给客户端	200(OK), 返回编辑文档及相关信息, 如更改时间, 大小等;
PUT	接收上传的 OFFICE 文档内容并保存	200(OK);
HEAD	得到 OFFICE 文档元信息 (如文档类型, 或保存痕迹等)	200(OK), 返回与编辑文档相关的信息, 如文档大小等;
LOCK	修改是否锁定的标志位	1. 200(OK), 锁定成功 2. 423 locked, 请求资源已被锁定
UNLOCK	修改是否锁定的标志位	200(OK)

3 在线读写在 .net 下的实现

IIS 5.0 以上内置提供了对 WebDAV 协议的基本支持。IIS 实现 WebDAV 协议是采用的其两种接口 CGI、ISAPI 的 ISAPI 接口。但其没有采用映射的方式，所以 IIS 的主程序 w3svc.dll 本身包含了 WebDAV 的信息，其识别出是 WebDAV 的请求后就调用 WebDAV 的处理模块 httpext.dll。

为了获得程序更好的扩展和安全性，通常情况下会将文档存入数据库中进行管理，故不直接把

WebDAV 协议请求交给 IIS 进行处理。也即需重写部分 WebDAV 协议的请求处理及响应。例如，默认情况下 IIS 收到 PUT 请求并上传文档存储于文件系统；重写 PUT 请求后，程序接管此请求，将上传的文档存储于数据库中。相应地，在接收到 LOCK、UNLOCK 及 HEAD 等请求时，程序将从数据库中找到文档的状态信息等作用响应。完成了服务器端请求、响应的处理，则实现了系统主要功能。

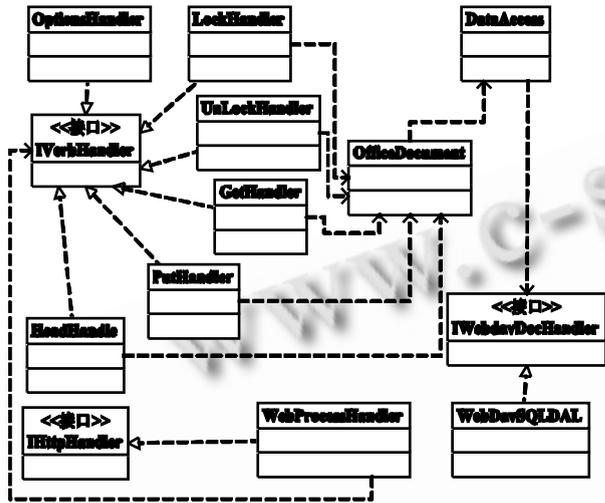


图 2 系统实现类图

上图为系统服务器端实现的类设计图，设计方案中预留了相应接口可以提供不同数据库的实现。其中 3 个接口：接口 IHttpHandler 是 .netFrameWork 用于实现自定义 HTTP 处理程序的接口，WebProcessHandler 实现此接口并在 web.config 文件中对请求配置进行重新定义即可实现自定义的 HTTP 协议请求操作；接口 IVerbHandler 用于实现 WebDAV 自定义请求，类 LockHandler、UnLockHandler、GetHandler、PutHandler、HeadHandler 及 OptionsHandler 分别实现此接口以处理不同的请求；接口 IWebDavHandler 用于实现 OFFICE 文档数据库读写，只需实现此接口，就可以实现对不同数据库的操作。DataAccess 为静态工厂类，OfficeDocument 类用于实现对 OFFICE 文档的具体读写操作。

为了在数据库中存放 OFFICE 文档及其相关信息，至少需要一张表。如果需要实现痕迹保留的话，还需要更多的表。本文中以 MS SQL 2000/2005 为例，表结构如表 2 所示：

表 2 用于存放文件信息及内容的表

名称	类型及长度	用途
FileID	char(16), not null	文件的唯一标识
FileName	varchar(256), not null	文件名称
CreatorName	nvarchar(256), not null	文件创建者名称
LastModifyName	nvarchar(256), not null	最后修改者名称
LastModifyDate	datetime, not null	最后修改日期
FileSize	bigint, not null	文件大小
Content	image, not null	OFFICE 文件
isLock	bit, not null	文件是否被锁定

如表 2 所示，文件内容以二进制存放。MS SQL 2000/2005 支持对以 image 字段存放的 OFFICE 及 PDF 类型文件进行全文检索，对程序进行适当扩展后，还可以对这两种类型的文件进行全文检索。

3.1 锁定及解锁

WebDAV 协议旨在服务器上处理资源提供更多方法。这些附加的方法通常要求大量信息与请求和响应相关联，以便显式定义客户端或服务器的意图。通过 HTTP 协议传递所有信息的方法只是标头在请求和响应方面的责任，很难对请求中的多个资源应用标头信息，而且难于描述层次结构。因此 WebDAV 协议允许使用 XML 放在消息体来传输信息，允许客户端在 WebDAV 服务器上描述和设置属性。这些属性可用于对服务器上的资源进行索引、搜索和处理。因为 XML 本来就具有可扩展性，所以这些属性的类型和用途几乎是无限的。在 IETF RFC 2518 的定义中，LOCK、UNLOCK 请求的附加信息，如授权令牌、超时时间等都可以放在 XML 文件中进行描述。

以下是 OFFICE 向服务器端第一次发出的锁定请求，其中 HTTP 头仅给出主要的信息，省略处用“... ..”表示：

```

LOCK
/Web/Document/(2008)53b2010019/(2008)5
3b2010019.doc HTTP/1.1
... ..
<?xml version="1.0" encoding="UTF-8"?>
<lockinfo xmlns="DAV:">
//lockinfo 属性指定客户端需要建立的锁定类型
<locktype><write/></locktype>
    
```

```
//locktype 属性指定锁定类型: 写锁定
<lockscope><exclusive/></lockscope>
//lockscope 属性指定锁定机制
<owner>administrator </owner>
//owner 属性指定文件所有者
</lockinfo>
```

如以上代码所示, 在实际获取 OFFICE 文档之前, OFFICE 客户端会发送一个 LOCK 请求至服务器, 获得对资源的独占。“Translate: f”标头的作用是使 WebDAV 协议请求与 HTTP 协议请求相区别。由于 WebDAV 协议也使用 GET、PUT 等请求, “Translate: f”标头的作用即通知服务器端该请求是一个 WebDAV 协议的请求而非 HTTP 协议请求。“Depth: 0”为深度标头, 它指定应当针对哪些资源应用此方法。在本次请求中深度为“0”, 即仅只要请锁定请求的文档。

消息体中是一个用于描述 WebDAV 请求属性的 XML 文件。lockinfo 属性指定客户端需要建立的锁定类型。locktype 属性指定了锁定方式, 目前 WebDAV 仅定义了一种锁定的方式, 即写锁定: <write/>。<lockscope>指明了锁定的机制。WebDAV 定义了两种锁定机制: 互斥锁定(exclusive)和共享锁定(share)。互斥锁定是仅对单个用户授予资源访问权限的锁定。只有删除(UNLOCK)了对某个资源的互斥锁定后, 其它用户才能访问该资源。共享锁定允许多个用户接收资源锁定, 任何具有访问权限的用户都可以获得锁定。<owner>属性用于描述文档所有人, OFFICE 客户端会自动使用登录 Windows 用户的名称来填写此属性。

对于 LOCK 请求, 服务器端还需要一个唯一标识符来区别对于不同文档的锁定请求。在 WebDAV 中使用 opaquelocktoken URI 方案(opaquelocktoken URI scheme), 这是一个授权修改令牌。在用户通过身份认证后, 使用这个唯一标识的授权令牌还可保证同一用户不会不小心在另一地点对资源进行修改。如请求锁定资源被成功锁定, 则服务器端返回“200 OK”, 同时返回的还有一个授权修改令牌 opaquelock token。如请求锁定资源已经被锁定, 则服务器端返回“423 locked”, 返回的 OFFICE 文档为只读状态。

锁定成功之后, 客户端将发送 GET 请求, 以获取文档实体。之后, 客户端将持授权令牌每隔一段固定

时间对此资源做出一次锁定请求。经实际测试, 由 ActiveX 控件打开的 OFFICE 2003/2007 客户端并不理会服务器端给出的锁定超时时间, 而是固定地每 2 分 15 秒发出一次锁定请求。当客户端打开的文档关闭后, 将再发送一个 UNLOCK 请求, 请求服务器端解除对此资源的锁定。由于篇幅关系, 这里不给出 UNLOCK 请求及响应的实例。

服务器端除返回以上锁定成功或者失败的响应之外, 还必须处理防止由于资源未能解锁而造成永久锁定问题。例如客户端崩溃而未能发送 UNLOCK 请求, 资源即可能永久被锁定而不能被其它用户访问。如前所述, 由于客户端会定期发送 LOCK 请求, 解决问题只需在服务器端为 LOCK 设置超时时间即可。当 LOCK 超时而未接到新的 LOCK 或 UNLOCK 请求时, 系统将自动为资源将解锁。

3.2 读写

涉及读写的请求主要由对 GET 及 PUT 请求的处理实现。HTTP 协议也定义了 GET、PUT 请求, 因此必须对这两个请求的响应进行重写。OFFICE 2003/2007 客户端发出的 GET 请求如下所示:

```
GET
/Web/Document/(2008)53b2010019/(2008)5
3b2010019.doc HTTP/1.1
... ..
If:
(<opaquelocktoken:2255ddec-0f15-46f5-a
f07-842efc0f972e:633703909787500000>)
Translate: f
```

可以看到, 客户端使用 lock 请求返回的授权令牌对资源对行请求。服务器端接到请求后即返回 200 OK, 并在消息体中附加 OFFICE 文档, OFFICE 2003/2007 客户端在接收到响应后会自动将文档打开。

在 OFFICE 中进行保存操作时, 客户端同样执授权令牌向服务器端发送 PUT 请求, 并在请求消息体中发送 OFFICE 文档。保存成功后, 服务器端返回“200 OK”响应。由于实现方式大同小异, 这里不再一一给出 HEAD、OPTIONS 等请求和响应的实例。

3.3 权限控制

微软提供的 ActiveX 控件可以控制 OFFICE 客户端与服务器端始终工作在 WebDAV 协议下, 但此控件只支持与 IIS 默认的 WebDAV 协议方式工作。而 IIS

默认工作方式仅支持基本身份验证、分布式的密码验证访问(DPA)或者匿名访问,它可以在域控服务器下很好地工作,但不支持以 **cookies** 的方式进行身份认证。这也就意味着如果使用该 **ActiveX** 控件访问进行访问,就不能使用 **WebForm** 的方式进行身份验证。

解决此问题的一个办法是为 **WebDAV** 协议的应用另外建立第二个网站,并做出相应的身份验证配置,例如分配域帐号,使用 **IIS** 的基本身份验证等方式进行身份验证。但此方案的缺点是很难与 **B/S** 应用系统的身份验证整合。

另一个解决方案是先使用户通过身份验证,持身份验证再取得检索文档的唯一 **GUID** 来获取文档。在 **OFFICE** 客户端向服务器端发出请求之前,先向服务器端进行身份验证请求---为取得更好的用户体验,可采用异步方式进行请求。由于此时的请求是在浏览器进程内发出的,因此可以进行基于 **cookies** 的身份验证。在通过身份验证后,服务器端为请求的文档临时生成一个 **GUID**,并将此 **GUID** 作为该文档在数据库中的检索条件,同时将此 **GUID** 发送给客户端。客户端在接收到此 **GUID** 之后,再使用 **ActiveX** 控件将此 **GUID** 作为检索条件打开服务器端的文档,拥有只读

权限的用户则直接由数据库中获得该 **GUID**。当文档解除锁定后,该 **GUID** 自动被销毁。

4 总结

Web 已经成为 **Internet** 的基础,**HTTP** 协议则成为 **Web** 应用最广泛的协议。但 **HTTP** 协议存在一些缺点,从而限制了它作为综合的 **Internet** 通讯协议而被采用,尤其是在文档处理方面,不能提供以足够复杂的功能处理文档。**WebDAV** 协议提供了对 **HTTP** 协议的扩展,使应用程序可直接对 **Web Server** 直接读写,并支持文件锁定及解锁,还可以支持文件的版本控制。本文基于 **WebDAV** 协议提供了一种 **OFFICE** 客户端直接读写服务器上的 **OFFICE** 文档的方法,重写了 **WebDAV** 协议部份请求及响应,实现了读、写、锁定、解锁等功能。

参考文献

- 1 江雨燕.Web 环境下的在线协同编辑系统设计与实现.电子科技大学学报,2002,31(6):630-635.
- 2 MSDN:SharePoint Team Services Client API:Client API Reference

