

一种 Web 视频监控播放插件^①

汪 畅, 袁怀月

(中冶集团 武汉勘察研究院有限公司, 武汉 430080)

摘 要: 针对不同设备厂商的视频不能在统一界面集中监控的问题, 设计了一种 Web 视频监控播放插件, 该插件根据网络配置与前端视频设备直连通信, 集成不同视频设备厂商提供的 API, 实现不同厂商设备的兼容. 基于 ActiveX 和 NPAPI 的插件实现机制, 开发了实时预览、历史录像下载与回放和本地录像文件播放等功能接口, 为不同浏览器的使用者提供相同的远程视频监控体验. 该插件在山东润兴化工智慧园区建设项目中得到了成功应用.

关键词: 视频监控; ActiveX; NPAPI; 浏览器插件; 兼容性

Web Video Surveillance Plugin

WANG Chang, YUAN Huai-Yue

(Wuhan Surveying-Geotechnical Research Institute Co., Ltd. of MCC, Wuhan 430080, China)

Abstract: For watching videos from different equipment manufacturers on the unified interface, a web video surveillance plugin is designed. According to the network configuration, the plugin is connected with the video device directly. Based on integration of the APIs which are provided by different video equipment manufacturers, the plugin is compatible with video devices from different manufacturers. Based on the mechanism of ActiveX and NPAPI, the plugin interface is developed. Through the plugin in different browsers, we can watch the real-time video and the local video, download and play the remote historical video. It provides the same remote video surveillance experience for users. The plugin has been successfully applied in the wisdom of park construction project of Shandong Runxing Chemical Technology Co., Ltd.

Key words: video surveillance; ActiveX; NPAPI; browser plugin; compatibility

由于各监控系统分离导致信息孤岛现象, 对安全监控关注度较高的企业(比如石油、化工、钢铁、能源、电力企业等)越来越倾向于综合性一体化的监控管理系统. 作为安全监控的重要组成部分, 视频监控被嵌入到综合系统中, 这就对视频监控嵌入形式提出了更高的要求: ①不同视频设备厂商提供的视频监控系统不一致造成不同位置的监控视频只能在不同的系统中查看, 因此, 在统一的界面进行视频监控显得非常有必要; ②人们对桌面应用程序的要求越来越趋向于 Web 化, 即在浏览网页时便捷查看视频, 免去下载并安装大文件的过程; ③用户会用不同的浏览器对现场视频进行查看.

由于视频图像数据量较大, 传输视频数据需要较大的网络带宽, 通常情况下, 监控视频在局域网或者企业专网中传输, 而随着网络带宽的加大, 通过互联网观看实时监控视频成为了可能. 王羽^[1]设计并实现了一种视频监控系统, 视频通过 HTML5 中 video 元素的 src 属性链接播放; 陈晓伟^[2]、徐佳健^[3]、王亮^[4]等均基于 ActiveX 设计了视频监控插件; 李迪等^[5]设计并实现了基于 QtWebkit 的浏览器视频插件; WANG Yongli^[6]设计了分布式视频监控系统架构; XIAO Donghui 等^[7]研究了集成视频监控平台解决方案. 本文在分析主流浏览器插件机制^[8,9]的基础上设计并实现一种视频监控播放插件. 与其他方法相比, 该插件兼

① 收稿时间:2016-01-19;收到修改稿时间:2016-03-01 [doi: 10.15888/j.cnki.csa.005335]

容主流的浏览器,同时,兼容不同厂商的视频设备.此外,为减小视频的时间延迟,该插件直接与前端视频设备进行通信.

1 浏览器插件介绍

当前,人们经常使用的浏览器有 Internet Explorer (Trident 内核,也称其为 IE 内核)、Google Chrome (WebKit 内核)、Mozilla Firefox(Gecko 内核)、Apple Safari (WebKit 内核)、Opera(Presto 内核)、360 浏览器 (Trident 内核、WebKit 内核)、搜狗浏览器(Trident 内核、WebKit 内核)等等,各主流浏览器的内核引擎为 IE、Gecko、WebKit、Presto 中一种或两种,其中,IE 内核支持的插件通过开发 ActiveX 插件实现,非 IE 内核支持的插件通过开发 NPAPI 插件实现.

ActiveX 是微软提出了一项 COM 组件技术,它的最大特点在于可以被基于 windows 平台开发的大多数应用程序所嵌入,应用程序作为组件的容器. ActiveX 控件是 ActiveX 技术的一部分,它以二进制文件(ocx 格式)的形式存在,ocx 是动态链接库文件格式中的一种,二进制文件形式这一特点,使得 ActiveX 控件具有与编程语言无关的特性,Internet Explorer(简称 IE)是微软公司 windows 操作系统自带的浏览器,因此,ActiveX 控件可以以插件的形式嵌入到 IE 中,应用于 web 应用程序,强大 web 应用程序的功能.

NPAPI 是 Netscape Plugin Application Programming Interface(网景插件应用程序编程接口)的缩写,Gecko 内核、WebKit 内核、Presto 内核均符合 NPAPI 接口开发标准,所以,利用 NPAPI 标准开发的插件可以嵌入在 Mozilla、Firefox、Google Chrome、Apple Safari、Opera 等浏览器页面中使用.

2 总体框架设计

Web 视频监控系统主要包括 Web 服务器、浏览器客户端、前端设备和视频监控播放插件.前端设备负责视频信号数据的采集、压缩编码并将数据打包发送到网络上;Web 服务器负责应用程序或网站的部署和发布;浏览器客户端通过网络地址向 Web 服务器请求访问监控页面,监控页面主要包括 html 页面元素、CSS 样式和 JavaScript(简称 JS)交互程序;浏览器客户端向 Web 服务器请求下载视频监控播放插件,插件需在本机上注册后才能加载在监控页面中,插件通过网络配

置直接与前端视频监控设备连接并通信,浏览器客户端通过与插件的 JS 通信实现视频的播放与控制.

考虑到视频信号采集过程、Web 浏览机理以及插件通信机制,设计了视频监控播放插件总体框架,虚线框部分为插件的主体部分,如图 1.

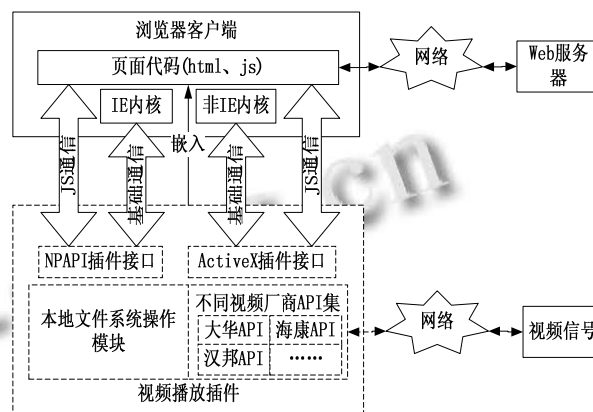


图 1 视频监控播放插件总体框架

插件包括四个部分:本地文件系统操作模块、视频 API 集成模块、NPAPI 插件接口和 ActiveX 插件接口.其中,本地文件操作模块负责加载与设置本地配置文件、视频截图与录像存储以及本地文件的读取.视频 API 集成模块是连接视频监控播放插件与前端视频设备的桥梁,也是视频监控播放插件最重要的组成部分.通常情况下,视频设备厂商都提供软件二次开发包 SDK,开发接口文件大多形式为动态链接库文件(.dll 格式文件),而采用面向对象语言 C++调用 dll 文件是最便利、最直接的方式,因此,为了保持统一、提高开发效率,插件的整个开发过程选用 C++语言. NPAPI 插件接口和 ActiveX 插件接口是连接浏览器客户端和视频 API 集成模块的桥梁,接口函数直接决定了用户能够使用的视频功能.插件在建立了与浏览器基础通信后,开发与浏览器的 JS 通信功能,实现浏览器与前端视频设备的间接通信.由于 NPAPI 插件和 ActiveX 插件实现机制的不同,这两部分的开发工作分开进行,但两部分均基于同一套视频设备厂商提供的 SDK 开发相同的功能,充分保证一致的用户体验.

3 模块设计与实现

3.1 视频集成模块

视频信号采集与编码设备分为两类:一类是模拟摄像机和视频服务器或编码器;另一类为网络摄像机,

它可以看作是由摄像机和视频编码设备组成的设备。在数据组织形式的设计中，把视频服务器或编码器定义为视频设备，把摄像机定义为视频设备的一个通道，简称视频通道。在此基础上，以“视频设备-视频通道”两级目录的方式组织视频配置数据，并进行了类图设计，如图 2 所示。

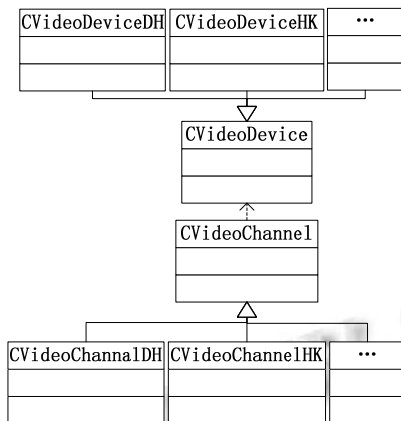


图 2 不同视频设备厂商 API 集成开发类图

对图 2 进行如下说明:

① 不同视频设备厂商的视频设备类继承视频设备类 CVideoDevice, 如海康威视的设备类 CVideoDeviceHK、大华的设备类 CVideoDeviceDH 等。不同视频设备厂商的视频通道类继承视频通道类 CVideoChannel, 如海康威视的设备通道类 CVideoChannelHK、大华的设备通道类 CVideoChannelDH 等。

② 视频设备类 CVideoDevice 的子类和视频通道类 CVideoChannel 的子类均实现了相应视频设备厂商提供的 API 接口, 根据面向对象设计的多态机制, 子类均采用反射机制进行实例化。

③ 视频设备类 CVideoDevice 与视频通道类 CVideoChannel 是依赖关系, 视频通道类的实例化需要有相对应的视频设备对象参数传入。

④ CVideoDevice 类及其子类负责视频设备的登录登出及设备参数的远程设置。

⑤ CVideoChannel 类及其子类负责单个视频通道的播放与控制, 包含实时预览、远程回放、录像下载、截图、语音控制、云台控制等, 其中, 录像下载和截图等功能涉及到与本地文件系统的交互。

插件主要实现了三部分的视频功能: 实时预览功

能、本地录像文件播放功能和历史录像回放与下载功能。其中, 实时预览功能和历史录像回訪与下载功能主要通过调用厂商提供的网络库和软解码库的接口函数实现, 本地录像文件播放功能主要通过调用厂商提供的软解码库的接口函数实现。三部分功能实现的流程图如图 3 所示。图 3 中虚线框部分由 CVideoDevice 类及其子类负责实现, 实现框部分由 CVideoChannel 类及其子类负责实现。

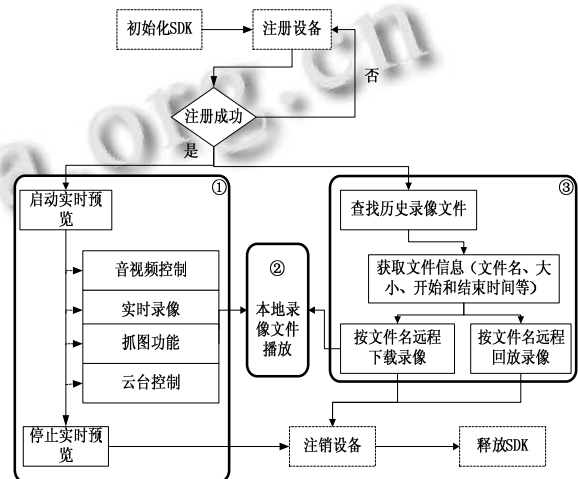


图 3 视频功能实现流程图

3.2 NPAPI 插件接口

NPAPI 标准提供了一套 JS 与插件的交互方法。这里主要介绍下 JavaScript 如何调用插件方法。

浏览器首先调用 NPError NPP_GetValue(NPP instance, NPVariable variable, void* value) 取得 NPObject 对象的地址, variable 参数为 NPPVpluginScriptableNPObject, 在取得 NPObject 对象后浏览器就可以调用插件提供的 NPClass 函数。插件关于方法调用的函数如下:

① pluginHasMethod: 插件是否支持某一 JS 方法。

② pluginInvoke: 当插件支持某一 JS 方法时, 插件将会调用该函数, 在该函数内部匹配并实现 JS 方法。插件如何在该函数内区分不同的方法呢, 分析这个函数:

```

bool pluginInvoke(NPObject *obj, NPIdentifier name, const NPVariant *args, uint32_t argCount, NPVariant *result);
    
```

其中, obj 是插件里的 NPObject 对象地址; name 表示插件提供方法名称的 NPIdentifier 对象, 通过对比这个参数来区分插件提供的不同方法; args 和 argcount 分

别表示 JavaScript 函数的参数地址和参数个数; result 是 JavaScript 函数的返回值地址。

以播放实时视频功能为例, 声明一个 NPIdentifier 对象 PlayReal 并初始化 PlayReal = NPN_GetStringIdentifier("PlayReal")。若包含 PlayReal 方法, 则 pluginHasMethod 返回真, 调用 pluginInvoke 触发具体功能的实现; 否则返回假。

3.3 ActiveX 插件接口

本文使用 VS2010 集成开发平台创建 <MFC ActiveX 控件> 工程项目, 工程创建好后 XCtrl.cpp 文件中相应的 GUID 应对应网页中控件的 classid, 实现网页与 ActiveX 控件的连接。

以播放实时视频功能为例, 在 X.idl 文件中, 声明 ActiveX 控件外部调用方法, 如下:

```
[id(3)] VARIANT_BOOL PlayReal(LONG lChannelNO);
```

COleControl 的继承类 CXCtrl 实现 ActiveX 控件内部的方法, 如下:

```
VARIANT_BOOL FirePlayReal(LONG lChannelNO);
```

控件外部调用方法 PlayReal 与控件内部方法 FirePlayReal 由调度映射建立联系, 如下:

```
BEGIN_DISPATCH_MAP(CXCtrl, COleControl)  
DISP_FUNCTION_ID(CXCtrl, "PlayReal",  
dispidPlayReal, FirePlayReal, VT_BOOL, VTS_I4)
```

```
END_DISPATCH_MAP()
```

其中, dispidPlayReal 是枚举类型, 值为 3, 与 X.idl 文件中的 id 值对应; VT_BOOL 为返回值, VTS_I4 为参数列表中的第一个参数类型, 两个以上的参数类型用空格分开。FirePlayReal 函数实现具体的播放实时视频功能。

4 实际应用

在网页中加载视频监控播放插件采用两种形式, 其中, 在 IE 内核浏览器页面中, 插件采用 html 中 object 元素的形式, 具体形式为:

```
<object id="VideoX" classid="CLSID:9DD32905-1072-4AD9-BDA2-E12EC717CCC3"></object>;
```

在非 IE 内核浏览器页面中采用 html 中 embed 元素的形式, 具体形式为:

```
<embed id="VideoX" type="application/x-wk-video"></embed>.
```

其中, 浏览器内核通过 navigator 对象的信息进行判断; id 属性是元素的唯一标识; classid 属性是 ActiveX 插件的唯一标识, 后面的内容是注册表里的信息; type 属性为 MIMEType, 它是利用 NPAPI 标准开发的插件的唯一标识。浏览器客户端通过唯一标识找到视频监控播放插件。

获取插件对象的 JavaScript 代码为 VideoPlugin = document.getElementById("VideoX"), 如果需要调用该插件实现一个具体的功能, 如播放实时视频, JavaScript 代码为 VideoPlugin.PlayReal(1), 然后通过插件内部的方法来实现具体的功能。

5 结语

本文设计了一种 web 视频监控播放插件。视频实时性强对安全性关注度较高的企业是非常有必要的, 因此, 采用插件与视频前端设备直连通信, 视频编码数据在客户端进行软解码, 避免视频信号延迟过大。采用软件设计模式, 集成了不同视频设备厂商提供的 API, 兼容不同厂商视频设备, 开发了实时预览、历史录像下载与回放和本地录像文件播放等功能。此外, 分别开发了 ActiveX 插件接口和 NPAPI 插件接口, 为不同浏览器的使用者提供相同的远程视频监控体验。

参考文献

- 1 王羽, 杨文娟, 陈侃. 基于 WEB 的跨平台移动视频监控系统. 科技创新导报, 2014, (17): 55-56.
- 2 陈晓伟, 孟利民. 基于 B/S 架构的视频监控音视频模块的设计. 电声技术, 2014, 38(6): 77-80.
- 3 徐佳健. 基于 ActiveX 的网络视频监控客户端设计[硕士学位论文]. 杭州: 浙江工业大学, 2013.
- 4 王亮. 基于门户服务器的视频监控系统的研究与实现[硕士学位论文]. 西安: 西安电子科技大学, 2008.
- 5 李迪, 丁娅. 基于 QtWebkit 的浏览器视频插件的设计与实现. 计算机技术与发展, 2012, 22(2): 238-240.
- 6 Wang YL. Architecture of distributed networked video monitoring system. Process Automation Instrumentation, 2013, 37(5): 74-80.
- 7 Xiao DH, Lin L. Solution scheme of integrated video surveillance platform of power system. Automation of Electric Power Systems, 2013, 37(5): 74-80.
- 8 邹强. 基于 NPAPI 的移动应用浏览器插件的设计与实现[硕士学位论文]. 西安: 西安电子科技大学, 2011.
- 9 张静, 廖武陵, 张晓春. 用于全景图像浏览的 ActiveX 控件设计与实现. 矿冶, 2013, 22(11): 222-225.