

# Web 网页控件的智能识别及其自动化操作<sup>①</sup>

## Intelligent Discernment and Automatic Manipulation of Web-Page Controls

张 金<sup>1,2</sup> 刘小飞<sup>1</sup> (1.华中科技大学机械科学与工程学院 湖北 武汉 430074;

2.武汉开目信息技术有限责任公司 湖北 武汉 430223)

**摘要:** 互联网软件使用中常常遇到因为网络瞬间断开或提交失败,使得先前录入的繁杂表单数据需要重新填写;由于缺少实时交互性提醒,需要不断查询是否有最新电子邮件或通告等信息。本文为解决这类单调重复的手工操作,通过分析 Web 页面的程序结构,利用微软提供的二次开发技术,提出了一种基于 Web 页面的控件智能识别方法,实现其自动化操作。并通过应用实例,验证了该技术能够较好地解决此类问题,减轻人工重复劳动,提高自动化水平。

**关键词:** 自动化 接口 集合 模拟 web 页面 网页文档对象

### 1 引言

随着网络技术和信息技术的飞速发展,各种 Web 开发平台的不断涌现,致使基于网络的 Web 应用软件迅速风靡全球,并渗入到社会的各行各业中<sup>[1]</sup>。原先基于桌面的办公软件、财务软件等都增添了网络应用,各种制造业企业的在线生产管理系统更是以 Web 页面的形式呈现在用户面前。人们在享受更多基于 Web 应用软件所带来的各种便利的同时,不得不面对众多 Web 页面重复批量手工操作的困扰,如日常的收发电子邮件、各种注册表单的填写、最新数据批量检索、频繁的 Web 页面数据提交等,常常会消磨掉人们的耐心,增加误操作率,降低工作效率。本文试图研究某种技术手段,针对这些机械重复的操作,开发出拥有相应流程的辅助操作软件,来代替人工烦琐的操作。

对此,可以有两种解决方案:(1)利用 Web 应用软件提供的二次开发接口;(2)研究类似于 Windows 系统中的 API HOOK 技术,在无二次开发接口的情况下对 Web 页面的控件进行捕捉,监控其执行。

由于 Web 应用软件千姿百态,提供的二次开发接口也各不相同,对于开发人员来说,每针对一种 Web

应用软件,都要学习其二次开发接口来进行编程,这无疑增加了负担。因此,相对来说,第二种解决方案更有意义。本文着重研究了基于 Web 页面的控件智能识别及其自动化操作技术,下面简称为“Web 页面自动化操控技术”。

### 2 Web 页面自动化操控技术关键

自动化技术的关键在于能够用程序控制各种对象,执行人们预设定的动作。在软件领域,对象实体具体指界面上的各种控件。一个基于 Win32 的应用程序,启动后,会在桌面出现主窗口,每个窗口都有其句柄和指针,我们可以通过 Win32 API 函数来获取当前运行窗口的句柄,进而获取它的指针,通过该指针又可以获取到窗口中的各个子控件的句柄,并可对各个子控件发送消息,令子控件执行各种操作。

然而,对于基于 Web 的应用程序来说,只有它的外壳(即浏览器)是一个标准 Windows 窗口控件,浏览器所包含的 Web 页面中,各种子对象(如文本框、按钮、超级链接)不是标准 Windows 控件,无法通过传统的 Win32 API 函数来获取它们的指针。但是可以利

① 基金项目:国家 863 高技术研究发展计划项目(2006BAF01A44,2007AA040605)

收稿时间:2008-09-22

用微软在发布 Internet Explorer 的同时所提供的二次开发接口, 获取 Web 页面中各种对象的指针, 从而达到操纵页面对象的目的。

### 2.1 Web 页面的结构

IE 浏览器是采用 DOM(文档对象模型)来管理网页的数据。它通过一个容器 (IWebBrowser2/ IHTMLWindow2) 来装载网页文档 (IHTMLDocument2)。而一个页面文档, 又可以由若干个帧 (frame) 组成, 管理这些帧的接口叫“框架集 (IHTMLFramesCollection2)”。

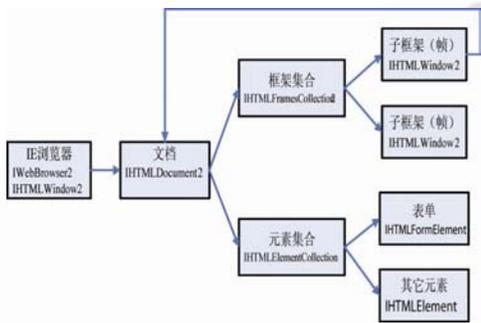


图 2 DOM 模型结构

对一个具体的网页来说, 其模型如下:



1.Document 2.Link(超链接元素) 3. InputElement(表单输入元素) 4. Figure(图片) 5. Frame(帧)

图 3 网页模型结构

### 2.2 Web 页面中各元素/控件的接口的获取

为了增强网页界面元素在客户端 IE 浏览器中的动态改变能力, 微软提供了一套自动化二次开发接口,

一方面可以在 ASP、JAVASCRIPT 脚本中操作网页界面元素, 另一方面也可供外部程序操纵网页界面元素使用。

有关浏览器自动化接口和 COM 对象如图 4 所示:

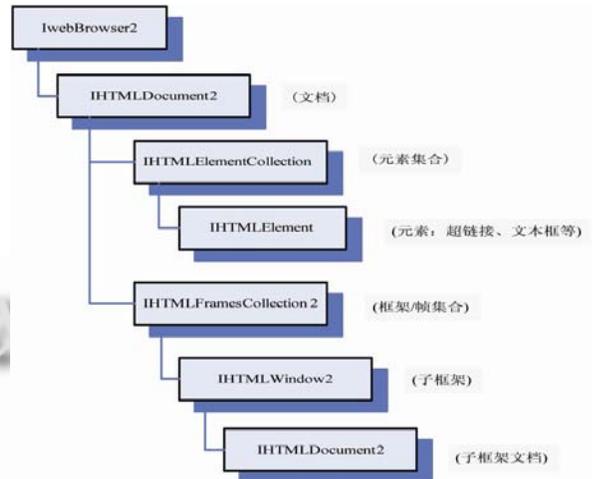


图 4 IE 浏览器自动化接口层次图

#### 2.2.1 网页文档对象(IHTMLDocument2)接口的获取

网页中的输入文本框、超链接等元素包含于文档对象(IHTMLDocument2)中, 要操纵这些元素, 必须先获取 IHTMLDocument2 对象的接口。根据 IE 浏览器的运行方式, 有六种不同方式可以获取文档接口[2]。分述如下:

##### (1) 使用 MFC 的 CHtmlView 类

如果在应用程序中使用 MFC 提供的 CHtmlView 类来浏览网页, 则可调用该类提供的函数来获取文档接口。该函数是 CHtmlView::GetHtmlDocument()。

##### (2) 使用 Web 浏览器的 ActiveX 控件

如果在应用程序中使用了 Web 浏览器提供的 ActiveX 控件, 同样可调用该浏览器提供的函数来获取文档接口。该函数是 CWebBrowser2::GetDocument()。

##### (3) 使用 ATL 写的 ActiveX 控件

如果在应用程序中使用了 ATL 写的 ActiveX 控件, 则需要调用成员函数 IOleClientSite::GetContainer 得到 IOleContainer 接口, 然后通过 QueryInterface() 查询得到 IHTMLDocument2 的接口。

##### (4) 使用 MFC 写的 ActiveX 控件

如果在应用程序中使用了 MFC 写的 ActiveX 控件, 则需要调用成员函数 COleControl::GetClientSite()

得到 `IOleContainer` 的接口,余下的步骤与(3)一样

#### (5) 取得独立运行着的 IE 浏览器的进程

如果 IE 浏览器作为独立的进程在运行,而每个运行着的浏览器都会在 `ShellWindows` 中进行登记,因此可以通过 `IShellWindows` 来取得浏览器实例,获得其 `IWebBrowser2` 接口,然后由 `IWebBrowser2` 提供的函数 `get_Document()`即可获得文档接口。

#### (6) 枚举浏览器窗口

如果 IE 浏览器被一个进程包装在一个子窗口中,可以先得到那个进程的顶层窗口句柄,然后枚举所有子窗口,通过判断窗口类名是否是 "Internet Explorer\_Server",从而得到浏览器的窗口句柄,再向窗口发送消息取得文档的接口。

### 2.2.2 网页各元素接口的获取

一个文档中的所有元素被装载到集合 `IHTMLElementCollection` 中,调 `IHTMLDocument2::get_all()` 取得元素集合,然后调用 `IHTMLElementCollection` 的 `length` 属性 `IHTMLElementCollection::get_length()` 得到元素的总数目,就可以循环调用 `IHTMLElementCollection::item()` 来取得每个元素的指针了。

从网页的 DOM 模型结构图中可以知道,一个文档对象中可以包含多个子框架,如果网页是多帧(有子框架)则还需要遍历所有的子框架。这些子框架 (`IHTMLWindow2`),被保存在集合中 (`IHTMLFramesCollection2`),利用函数 `IHTMLDocument2::get_frames()` 来获得框架的集合指针。然后取得子框架的总数 `IHTMLFramesCollection2::get_length()`,接着就可以循环调用 `IHTMLFramesCollection2::item()` 函数顺序地取得子框架 `IHTMLWindow2` 指针。通过 `IHTMLWindow2::get_document()` 获取文档对象接口后,余下的与单帧网页处理方式相同。

`IHTMLElementCollection::item()` 函数取得的是该集合中一个元素的 `IDispatch` 指针。如果它是一个超级链接或是按钮,要操作它则需要调用该元素的 `click` 方法。首先通过 `QueryInterface()` 查询,得到相应元素具体类型的指针。一般可以使用通用类型 `IHTMLElement` 来查询,这样可以调用 `click` 方法。如果它是一个文本框或密码框,则需要设置它的属性 `value` 值,可通过使用 `IDispatch` 包装类 ATL 智能指针: `CCoMDispatchDriver`。这个智能指针的好处就

是方便读取、设置属性值。如:

```
// 定义 IDispatch 的智能指针
CCoMDispatchDriver spInputElement;
//枚举
spElementCollection->item(CComVariant(j), CComVariant(), &spInputElement );
//定义元素名称, 值
CComVariant vName,vVal;
//读取名称
spInputElement.GetPropertyByName(L"name",&vName)
//设置其 value 属性
spInputElement.SetPropertyByName(L"value",&vVal );
```

### 3 Web自动化技术应用实例-自动查询并发电子邮件

日常电子邮件的查询,除少部分人使用微软公司提供的邮件客户端软件 Outlook 以外,大部分人习惯于直接登录邮件提供商的网站,输入用户名、密码后进入邮箱查询是否有新邮件。这个过程虽然单一,但企业业务可能需要不断检索是否有新邮件,故可以用 Web 自动化技术来完成。

对于上网查询电子邮件过程,人工操作的话基本只有键盘输入和鼠标点击。关键是人的眼睛发挥了无形的作用,只有找到输入框和超级链接的在网页中的具体位置,才可以有效地实施操作。这里,利用 Web 自动化技术,可以模拟人眼的作用,准确地捕获到 Web 页面上的文本输入框和超级链接的接口,然后对其实施必要的操作。我们可以模拟人工键盘输入、鼠标点击事件。在网页文本输入框中输入字符,可用“捕获网页上的输入框并置其 `value` 值”这一操作来代替;鼠标点击网页上的按钮或超级链接,可用“捕获网页上的按钮/超级链接元素的接口并调用其 `click()` 方法”这一操作来代替。过程的流程图如下:

由于是专门针对特定网页电子邮件的自动查询,在获取网页元素接口后,要给出判断条件并作出判断是否是目标元素,所以要事先获取网页元素的一些可见属性值。如元素名 `name`,标签名 `tagName`,类名 `className`,ID 号 `id` 等。使用微软提供的一款辅助网页设计工具 IE Developer Toolbar 可以查看各元素

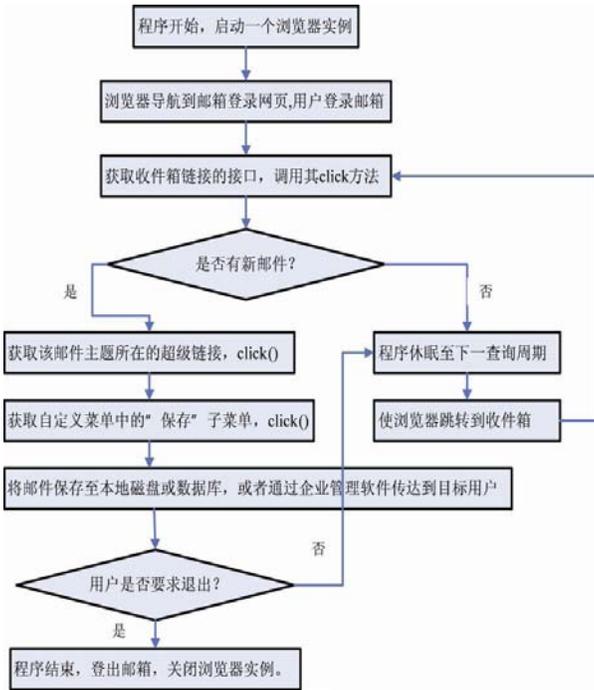


图 5 自动查询并分发电子邮件流程图

的属性值。在程序中用枚举手段对网页文档中的元素进行枚举, 通过 IDispatch 包装类 ATL 智能指针: CComDispatchDriver 来获取它的标签名 tagName 和 ID 号 id 等属性后, 再与事先用 IE Developer Toolbar 可以查到的元素相应属性值进行比较, 如果一致, 即找到该元素的接口(指针)。IE Developer Toolbar 是嵌入在浏览器中的, 方便查询各元素的可见属性。如图 6 所示。

找到目标对象后, 需要输入用户名和密码的, 可通过智能指针设置其属性 value:

splInputElement.PutPropertyByName( L"valu e",&vVal ); 对于需要调用其 click 方法的, 可通过 QueryInterface() 查询, 取得其 IHTMLElement 指针后, 再调用 click() 方法。



1. 被选中的按钮, 其属性显示在 IE Developer Toolbar - ar 中 2. IE Developer Toolbar 嵌入网页图示  
图 6 IE Developer Toolbar 嵌入网页图示

### 4 总结

通过分析 Web 页面的结构, 利用微软公司提供的一套针对 IE 浏览器的二次开发接口, 本文提出了基于 Web 页面的控件智能识别技术, 利用该技术, 可以针对特定的 Web 页面, 开发出专门的自动化软件代替重复、单调的批量手工操作。不仅可应用在自动收发电子邮件上, 还可根据用户需求, 设计出具有特色的软件, 如可操作搜索引擎自动搜集某一行业新闻、当网络突然断开时自动保存网页表单信息, 待网络恢复后可自动填写表单、自动执行 Web 页面软件的日常测试等。

### 参考文献

- 1 路晓丽, 葛伟, 郝克刚. Web 应用软件测试模型的研究和构建. 计算机科学, 2006, 33(1): 175 - 176.
- 2 管红杰, 孟凡荣. DOM 应用 - 遍历网页中的元素. 计算机工程与设计, 2006, 27(1): 99 - 101.