

# 基于VC++的Word文档自动生成技术

熊瑜容, 柴 毅, 王淑娟, 王嘉骐

(重庆大学自动化学院, 重庆 400044)

**摘 要:** 采用COM技术调用OLE自动化对象, 实现了Word文档在VC++平台中的自动生成。运用书签定位和光标跟随相结合的技术, 在自动生成的Word文档中精确插入了文字、图片、表格和目录。

**关键词:** VC++; COM技术; OLE自动化对象; 书签

## Word Document Automatically Generated Technology Based on VC++

XIONG Yu-rong, CHAI Yi, WANG Shu-juan, WANG Jia-qi

(College of Automation, Chongqing University, Chongqing 400044, China)

**Abstract:** Adopting COM technology to transfer OLE Automation objects, it is realized that word document is automatically generated on the VC++ platform. Using the technology of combining bookmark locating and cursor tracking, the characters, pictures, tables and catalog have been exactly inserted in the word document generated automatically.

**Key words:** VC++; COM technology; OLE Automation objects; bookmark

## 0 引言

各行各业每天都有大量的报告需要编制, 而大多数情况下仅仅只是其中的数据、图表不同而已, 如此周而复始的劳动, 依靠人工通过计算机编辑表格、录入数据、绘制曲线, 不仅劳动量巨大, 而且容易出错。因此, 规范准确的Word报告的自动生成具有重要的应用价值。

在报表的自动生成方面, 文献[1]主要采用VBA (Visual Basic for Application) 编程。VBA是实现Word文档自动生成的强有力的编程工具, 但在软件重用性与代码安全性上存在缺陷和漏洞, 不能满足继承和封装的要求, 不利于软件的进一步扩展。文献[2]主要采用Delphi来实现。Delphi与BDE (Borland Databas Engine) 的无缝集成, 以及Delphi提供的现成数据库操作控件, 使得Delphi在数据库支持上有着无可比拟的优势, 但其在兼容性和稳定性等方面存在着许多的不足。文献[3]介绍了用VC++与VBA实现复杂报表自动生成的方法。但是, 在日常工作中, 报告里不仅仅只有表格, 还有文字、图片等其他内容, 该方法无法满足这些要求。

本文采用VC++编程实现了文字、图表并茂的规范文档的自动生成。Word提供的强大的编程接口技术让我们能够通过各种编程工具使用Word。在VC++环境中, 我们将Microsoft Office安装目录中的MSWORD.OLB类库导入VC++中, 利用COM技术编程, 在源代码中调用OLE自动化对象来操作Word。鉴于报告文档的格式相对固定, 开发人员多采用事先设计好的模板文件, 在需插入内容的地方设置书签变量。所以, 文中通过书签定位和光标跟随相结合的方法来实现特定位置的内容插入, 以减少代码量, 提高程序执行率。

## 1 文档自动生成的原理

Word文档自动生成原理如图1所示。



图1 Word文档自动生成原理图

VC中搭建应用平台: 在VC中建一个具有文字、图片、表格的界面。

制作模板: 为了避免反复输入相同的信息或做重复的格式修改, 引入了模板技术。创建自定义模板的方法如下: 新建一个空白的Word文档, 在文档中添加相关信息, 设置好要求的格式, 将这些做为模板的静态文本。对于需要动态改变的部分, 则在相应的位置插入书签, 并设置书签变量。然后将所做的Word文档以dot的形式保存。这样就完成了模板的创建。

文字、图表等的插入: VC++通过COM技术调用Word自动化服务器中的OLE自动化对象, 如: \_Application、\_Document、Range、Selection、Bookmark、InlineShape等, 实现文字、图片、表格插入, 标题的设置以及目录的生成。

## 2 功能的实现

### 2.1 创建OLE自动化对象

(1) 在程序的应用程序类的InitInstance()成员函数中, 加入AfxOleInit()函数的调用, 初始化COM。

```
if (!AfxOleInit())
{
    AfxMessageBox(IDP_OLE_INIT_FAILED);
    return FALSE;
}
```

(2) 在程序导入Word中OLE自动化对象。主要是通过add class找到Office安装位置下的Word类型库文件MSWORD.OLB, 从中添加需要用到的对象。主要有: Application

对象、Document 对象、Range 对象、Selection 对象、Bookmark 对象。Word 对象模型是一个等级结构,如图 2 所示。

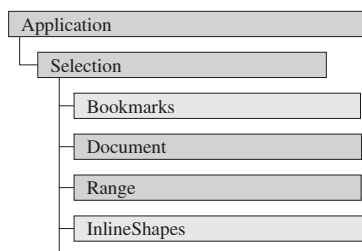


图 2 Word 对象模型

**Application 对象。**用户可以利用它的属性或方法来控制应用程序的运行行为、控制 Word 窗口的外观或调整 Word 对象模型的其他参数。

**Document 对象。**表示一个打开的文档,几乎所有的操作都要调用 Document 对象本身或其内容。用户可以使用 Document 对象或 Documents 集合的属性或方法来打开、创建、保存、启动或关闭文档。

**Range 对象。**表示文档中的一块连续的区域。每一个 Range 对象都是由一个起始字符位置和一个终止字符位置定义。在确定 Range 对象后,就能够应用该对象的方法和属性来修改区域内的内容或格式。

**Selection 对象。**当用户想要自己的代码对选定内容做出响应或是改变选定内容,就可以通过 Selection 对象来完成任务。Selection 对象可以是一个插入点,也可以是几个字符,或是文档的部分内容。用户可以用其方法、属性来移动或扩展选定内容,本文选用 GoTo 方法来移动选定内容。

**Bookmark 对象。**表示文档中的连续区域,既有起始位置也有结束位置。书签用于在文档中标记一个位置,或者用做文档中的文本容器。用户可以在文档中定义多个书签,把 Bookmark 看作是保存在文档中的一个指定位置。

## 2.2 创建 Word 服务及连接

前面介绍了文档生成需要用到的一些基本对象,VC 要调用这些对象,还必须得做如下工作:

首先要创建 Word 服务。代码如下:

```

if(!m_App.CreateDispatch("Word.Application"))
{
    AfxMessageBox("创建 Word 服务失败!");
    exit(1);
}
  
```

其次要将 Documents 类对象 m\_Docs 和 IDispatch 接口关联起来。IDispatch 是调度接口,是所有支持自动化的 COM 组件都必须实现的接口之一。

```
m_Docs.AttachDispatch(m_App.GetDocuments());
```

## 2.3 书签定位与光标跟随实现精确定位

在文献[4]中采用了光标定位的方法,在定位时有许多繁杂的计算。对于格式相对固定的文档来说,静态部分可以精确计算出字段的长度,但对于动态部分,由于每次插入字段数是不确定的,一旦字段数计算不准确就很容易造成文字的错位,这样不仅增加程序的复杂度,而且也会造成文档的混乱。本文主

要选用书签定位并结合光标跟随的方法,来实现文字、图片、表格、目录等的精确插入。通过书签变量的设置,在 Word 模版中能够准确地将要插入的内容插入于指定位置。而一段内容填写完毕后,光标就停留在最后操作的地方,要转到其他位置则采用光标跟随法,将光标转到另一书签处,然后实现图表等的插入。这样程序相对简易,而且文档标准美观。

为了便于后面的描述,将要用到的几个变量定义如下:

```

_Application m_App; //控制 Word 应用程序
Documents m_Docs; //操作所有的文档
Selection m_Sel; //当前所选内容
_Document m_doc; //操作某个文档
  
```

### 2.3.1 文字的插入

首先获得打开模版的书签集,然后将模版中的书签变量与应用界面上变量对应起来,最后将应用界面编辑框中的文字写进 Word 模版中的相应位置。主要代码如下:

(1) 获得 Word 书签集

```

Bookmarks m_bks;
Bookmark m_bk;
m_Doc=m_App.GetActiveDocument();
m_bks=m_Doc.GetBookmarks();
  
```

(2) Word 中的书签变量与对话框界面上的编辑框变量对应起来。

```

COleVariant bk_name("name"); //“name”为书签变量
m_bk=m_bks.Item(&bk_name);
m_Rng=m_bk.GetRange();
m_Rng.SetText(m_name); //“m_name”为编辑框变量
  
```

### 2.3.2 图片的插入

在此采用光标跟随的方法将光标转到需插入图片的书签变量处,然后利用 InlineShapes 函数来插入图片。

(1) 采用光标跟随法将光标转到 Word 模版中的图片书签变量“pic”处。

```

m_Sel.GoTo(COleVariant((short)-1),COleVariant((short)0),
COleVariant((short)0),COleVariant("pic"));
  
```

(2) 采用 InlineShapes 集合中 AddPicture 方法添加图片。

```

LPCTSTR FileName=m_pic;
// m_pic 为存有图片地址的编辑框变量
InlineShapes m_wdInlineShapes;
InlineShape m_wdInlineShape;
m_wdInlineShapes=m_Sel.GetInlineShapes();
m_wdInlineShape=m_wdInlineShapes.AddPicture(FileName,
vFalse,vTrue,vOpt);
  
```

### 2.3.3 表格的填写

(1) 规则表格的填写

因为规则表格格式相对固定,因此只需在表格的第一个单元格设置书签变量。用户可采用光标移动的方法,来填写表格内容。

```

m_Sel.GoTo(COleVariant((short)-1),COleVariant((short)0),
COleVariant((short)0),COleVariant("table"));
//转到 Word 模版里的表格书签变量“table”处
  
```

```
m_Sel.MoveRight(ColeVariant((short)1),ColeVariant((short)1),
    ColeVariant((short)0)); //采用 MoveRight 方法将光标移到下一
    单元格
```

```
m_Sel.TypeText(m_name); //填写单元格内容
```

## (2) 不规则表格的填写

不规则表格样式多变,主要采用书签定位的方法来填写,因此,只在需要填入数据的单元格设置书签变量,然后和编辑框变量对应。这样不仅插入精确,而且也不必为编号而费尽心思。

### 2.3.4 目录的生成

在报告内容量较大时,为便于查阅,有时要生成目录。在此给出设置标题以及生成目录的方法:

#### (1) 设置指定文字为标题

```
m_Sel.GoTo(ColeVariant((short)-1),ColeVariant((short)0),
    ColeVariant((short)0),ColeVariant("title"));
//转到 Word 模版里的需要设置成标题的书签变量“title”处
ColeVariant vStyle((long)-2); //-2 为一级标题,...,-10 为九级标题
m_Sel.SetStyle(&vStyle); //采用 SetStyle 方法设置标题
```

#### (2) 目录的生成

```
Range m_rng;
ColeVariant bk_list("list"); //"list" 为目录书签变量
m_bk=m_bks.Item(&bk_list);
m_rng =m_bk.GetRange();
```

(上接第 51 页)

核心代码如下:

```
Public OMNode getIndexFromWS(){
    private EndpointReference targetEPR=new EndpointReference
        ("http://202.210.211:8088/axis2/services/GetIndexService");
    OMFactory fac=OMAbstractFactory.getOMFactory();
    OMNamespace omNs=fac.createOMNamespace("http://
        202.210.211.192/", "index");
    OMElement method=fac.createOMEElement("getIndex",omNs);
    method.setText("getIndex");
    Options options=new Options();
    options.setTo(targetEPR);
    ServiceClient sender=new ServiceClient();
    sender.setOptions(options);
    OMNode result=sender.sendReceive(method);
}
```

该段代码主要是向服务器发送请求,然后服务器返回 OMNode 对象,客户端可以从 OMNode 对象中,还原出包含指标维度的 List。

至此,报表平台取得了包含指标或者维度信息的 List,可以把这个对象读取到 bean 中,或者使用 Javascript 直接在前台展现供用户选择。前台用户在选择好指标信息后,系统把它封装成 SOAP 消息递交到事实数据提供平台。事实数据提供平台根据 SOAP 消息中的指标维度相关信息,生成 SQL 语句,在事实表中查找出服务调用者需要的数据,或者生成相应的


```
TablesOfContents ts; //TablesOfContents 集合表示文档的目录
ts=m_doc.GetTablesOfContents();
```

```
ts.Add(m_rng,&UseHeadingStyles,&UpperHeadingLevel,
    &LowerHeadingLevel,&UseFields,&TableID,
    &RightAlignPageNumbers,&IncludePageNumbers,
    &AddedStyles,&UseHyperlinks,
    &HidePageNumbersInWeb,&UseOutlineLevels);
```

### 3 结束语

本文所介绍的在 VC++ 中调用 OLE 自动化对象来实现 Word 文档的自动生成技术,将书签技术与光标技术灵活地结合在一起,实现了文字、图片、规则表格与不规则表格的插入,以及标题的指定与目录的生成。鉴于 VC 的广泛应用,本文的文档自动生成技术具有广阔的应用前景。

#### 参考文献:

- [1] 金良铮,周文祥.WORD 试验报告的自动生成[J].中国测试技术,2007.33(4):112~115
- [2] 鲁保玉,杨新芳.用 Delphi 生成 Word 报告及动态结构表格[J].计算机应用与软件,2007.24(3):180~183
- [3] 朱敏,沈同奎,王学伟,周晓东.VC++ 与 VBA 结合实现复杂报表[J].计算机应用与软件,2005.22(2):42~43,101.
- [4] 叶明,张静.基于 C#.NET 的 Word 报告生成功能开发[J].计算机工程与应用,2008.44(9):104~107
- [5] 杨晓亮.Word VBA 高数排版范例应用[M].中国青年出版社,2005. 

文件,返回给服务调用者。至此,一个灵活报表的生成过程结束。

### 3 结束语

该灵活报表生成模型已应用于重庆电信 BI 系统的营销项目子模块中。为了便于设计、维护和实现构件重用,采用了 Web Services 技术,把它与其它子模块分离开来。

与传统报表生成模型相比,本文提出的灵活报表生成模型允许用户自由选择报表指标,指标选择十分灵活,根据需要生成不同特性的报表。同时利用 Web Services 实现构件重用,使成本更低,实时生成报表更快捷。

#### 参考文献:

- [1] 梁晓路,梁宇奇.Web Services 技术、架构和应用[M].电子工业出版社,2003.
- [2] 朱德利.SQL Server 2005 数据挖掘与商业智能完全解决方案[M].电子工业出版社,2007.
- [3] 许科峰等.基于 Web Services 的企业应用集成技术及实现[J].计算机应用,2004.24:3
- [4] 顾宁,刘家茂,梁晓路.Web Services 原理与研究实践[M].机械工业出版社,2006.
- [5] Michael J. A. Berry, Gordon S. L. inoff. Data Mining Techniques: for Marketing, Sales, and Customer Relationship Management[M]. BeiJing: China Machine Press, 2006. 