

文章编号:1004-1478(2011)03-0100-04

# 基于 JACOB 的 Java 通用 WordWriter 组件设计

金保华, 李金旭, 赵丽辉, 吕书林, 林青

(郑州轻工业学院 应急管理信息技术研发中心, 河南 郑州 450002)

**摘要:**为了在 Java 编程中很好地调用 COM 组件、易于操作 Word 文档,采用 CBSD 的开发思路,利用组件重组技术,设计并实现了基于 JACOB 的 Java 通用 WordWriter 组件.通过该组件使用 Java 语言可以方便地生成各种格式的 Word 文件,使程序更易读、易维护,从而缩短软件开发周期,降低开发成本.

**关键词:**COM 组件;WordWriter 组件;JACOB;Java;CBSD

**中图分类号:**TP311.5      **文献标志码:**A

## Design of Java general WordWriter component based on JACOB

JIN Bao-hua, LI Jin-xu, ZHAO Li-hui, LV Shu-lin, LIN Qing

(Emergency Mana. Infor. Tech. R & D Center, Zhengzhou Univ. of Light Ind., Zhengzhou 450002, China)

**Abstract:** In order to solve the problems about the use of COM components controls in Java development and operating Word document format in the Java program, a JACOB-based Java general WordWriter component was designed using CBSD idea and component recombination technology. This component can easily generate various word documents formats. In addition, it makes the program more readable, easy to maintain and shorten the software development cycle and reduce development costs.

**Key words:** COM component; WordWriter component; JACOB; Java; CBSD

## 0 引言

在实现复杂软件或者进行集成环境的开发或测试时,往往要与各种应用程序打交道.理想情况下,如果能用一种编程语言实现所有功能是很令人憧憬的,但由于遗留代码、语言的能力和厂商的互操作性的策略不同,这往往是一个不现实的目标.比如可以使用 C#, C++ 或 VB 等语言方便地操作 Word 文档,但是如果您是一个 Java 程序员,事情就不那么简单了.如果可以在 Java 中调用 COM 组件,

就可以充分利用 Java 技能和现有成熟的 COM 工具包,大大简化应用开发的过程.

随着 JACOB 的出现,这样的问题得到了一定程度的解决.但目前 JACOB 没有一个完整的通用的使用接口,只提供了基于 JNI 技术实现的 Variant, Dispatch 等基础接口<sup>[1]</sup>.而这些基础接口的方法和参数却不是通常 Java 编程者所熟悉的,使得编程人员同样难以使用这些接口;即使 Java 编程者熟悉以后,在开发的过程中程序开发人员也要书写大批功能类似的重复代码,这就又出现了一个在软件开发

收稿日期:2011-03-09

基金项目:郑州市科技局基金项目(10PTGG339—6)

作者简介:金保华(1966—),男,河南省郑州市人,郑州轻工业学院副教授,主要研究方向为人工智能、辅助决策支持系统.

中比较常见的一个问题:重复代码.

为了解决这类问题,本文在 JACOB 的基础之上提出 WordWriter 组件,以期设计一个通用的、简洁的 Java 调用 Word 的接口组件.

### 1 WordWriter 组件的设计

本文中组件采用 CBSD(component - based software development)的开发思路,把 JACOB 组件中对 Word 复杂的操作进行归纳总结,以抽象出通用的 WordWriter 组件.

#### 1.1 CBSD 开发方法

传统的软件开发过程是串行瀑布式、流水线的过程,而 CBSD 实现了分析、设计、类等多层次上的重用<sup>[2]</sup>,是并发进化式、不断升级完善的过程,引导软件开发从应用系统开发转变为应用系统集成<sup>[3]</sup>.具有松耦合特性的组件不但实现了页面级的软件复用,而且实现了系统级的软件复用<sup>[4]</sup>,不但缩短了系统的开发周期,相应地也增强了系统的可维护性与健壮性.图 1 显示了它们的区别.

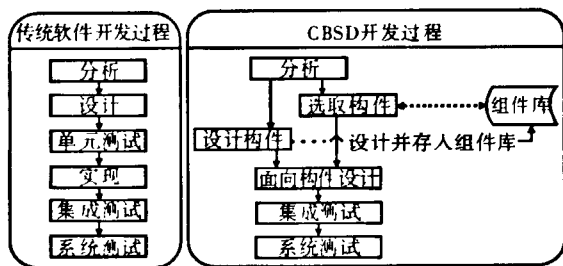


图 1 传统软件开发过程与 CBSD 开发过程比较

#### 1.2 JACOB 组件

JACOB 组件是 Java 和 COM 桥接的缩写,即 Java-COM Bridge<sup>[5]</sup>.通过使用 JACOB 类库,可以很方便地在 Java 程序中调用 COM 自动化组件. JACOB 最初是由美国人 Dan Adler 在“Inventre”公司担任 CTO 时编写的.目的是方便众多的程序员在 Java2 虚拟机上,调用 WIN32 平台上 COM 自动化服务器中的组件<sup>[6]</sup>.

JACOB 组件是一个 JVM 独立的自动化服务器实现<sup>[7]</sup>,其核心是基于 JNI 技术实现的 Variant, Dispatch 等接口.设计参考了 Microsoft VJ++ 内置的通用自动化服务器,但是 Microsoft 的实现仅仅支持自身的 JVM.通过 JACOB,可以方便地在 Java 语言中进行晚期绑定方式的调用.图 2 显示了 JACOB

的基本结构.

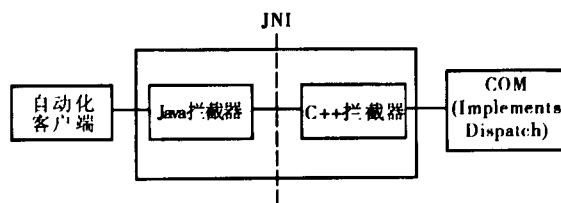


图 2 JACOB 基本结构

#### 1.3 WordWriter 组件的功能分析与工作流程

JACOB 组件中对 Word 文档的操作比较零散,需要用户逐条语句地书写.当用户下次需要设置相同的内容或格式时,仍需要将相同或类似的语句重复书写.这样就造成了程序代码的重复,使代码的可读性降低,给代码的维护造成了一定的困难.同时也必然会延长软件的开发周期,增加软件开发的成本.

WordWriter 组件基于 JACOB 组件将对 Word 文档进行控制的语句分类抽象出来,形成特定的方法接口.一般的 WordWriter 组件包含以下功能:

- 1) 创建 Word 文档.只需指定 Word 文档的名称和文档存放的路径,WordWriter 组件就会自动地创建出一个新的 Word 文件.
- 2) 页面设置.只需指定文档的上下左右的边距值,WordWriter 组件就会自动地将生成的 Word 文档设置为用户指定的格式.
- 3) 文字的设置.文字的设置包括文字的字体、字形、字号、字符间距等.
- 4) 段落的设置.段落的设置包括段前和段后的间距、行距等.
- 5) 插入表格.只需输入需要插入之表格的行数和列数,WordWriter 组件就会自动地生成有固定的行和列的表格.
- 6) 插入页码.用户可以选择页码的格式、页码的位置、首页是否显示等.
- 7) 其他的方法.如文字左对齐或居中、插入图片等.

WordWriter 组件的工作流程为: 1) 获得 WordWriter 对象实例; 2) 调用 WordWriter 对象实例的各种方法设置 Word 内容和格式; 3) 释放 WordWriter 对象实例.

其中 1) 和 3) 这 2 个步骤每次的操作完全相同,只有在步骤 2) 中需要用户根据不同的需求选择步

骤1)到7)中的不同方法.

## 2 WordWriter 组件的实现

在 JACOB 的基础上,对 Word 操作进行分析归纳,将实现特定功能的部分进行抽象,再进行适当的处理,即构造出 WordWriter 组件.具体的编码如下:

```

Public class WordWriter{
private Dispatch localWord = null;//Word 文档
private ActiveXComponent objWord = null;//Word 运行
程序对象
.....
//设置页边距
public void setPageSetup( int orientation, int leftMargin, int
rightMargin, int topMargin, int buttomMargin) {

```

```

if( this. pageSetup == null) {
this. getPageSetup( );
}
Dispatch. put( pageSetup, " Orientation", orientation );
Dispatch. put( pageSetup, " LeftMargin", leftMargin );
Dispatch. put( pageSetup, " RightMargin", rightMargin );
Dispatch. put( pageSetup, " TopMargin", topMargin );
Dispatch. put ( pageSetup, " BottomMargin", buttomMar-
gin);
}
.....
}

```

WordWriter 组件的核心属性和方法详见表 1, WordWriter 还有其他的一些方法和属性没有列出来.

表 1 WordWriter 组件的核心属性和公共方法

序号	名称	类型	数据类型	描述
1	objWord	属性	ActiveXcompenent	运行 Word 程序对象
2	localWord	属性	Dispatcher	指定一个本地的 Word 对象
3	documents	属性	Dispatcher	Word 的所有文档窗口( Word 是多文档应用程序)
4	document	属性	Dispatcher	一个新的 Word 文档
5	selection	属性	Dispatcher	代表窗口或窗格中的当前所选内容,如果文档没有选定任何内容,则代表插入点.每个文档窗格只能有 1 个 selection 对象,并且每个应用程序中只能有 1 个活动的 selection 对象.
6	pageSetup	属性	Dispatcher	包含文档所有页面设置的属性.如页面的左右上下页边距和纸张的大小.
7	WordWriter( )	方法	WordWriter	初始化 WordWriter 对象的一个实例
8	createNewWordDocument( )	方法	Dispatcher	创建一个新的文档.返回新建的 Word 文档
9	openWordDocument( String wordPath)	方法	Dispatcher	wordPath 为要打开的文件全路径.返回打开的文件.
10	getSelection( )	方法	Dispatcher	返回选定的范围或插入点
11	getPageSetup( )	方法	Dispatcher	返回当前文档页面属性对象
12	getRange( )	方法	Dispatcher	返回当前文档内可以修改的区域
13	createNewParagraph( )	方法	Void	在文档中插入一个新的段落
14	createNewTable( int rowCount, int colCount, int width)	方法	Dispatcher	rowCount 代表行数,colCount 代表列数,width 代表边框的值: 0 浅色,1 深色,返回新创建的表格对象
15	setAlignment( int type) setPageSetup( int orientation, int leftMargin, int rightMargin, int topMargin, int buttomMargin)	方法	Void	设置对齐方式 0 左,1 中,2 右 Orientation:可取值 0 或 1 分别代表横向或纵向,leftMargin:左边距的值,rightMargin:右边距的值,topMargin:上边距的值, buttomMargin:下边距的值
16	setPageNum( int type, int position)	方法	Void	设置页码,type:页码的类型;position:页码的位置(0 左,1 中,2 右)
17	insertContents( String contents)	方法	Void	在文档中插入文本内容,contents:插入的内容
18	insertImage( String imagePath)	方法	Void	在当前插入点插入一张图片,iamePath:图片的全路径
19	replaceAll( String oldText, String newText)	方法	Void	将文档中的 oldText 全部替换为 newText
20	saveAs( String outputPath)	方法	Void	保存文件 outputPath:保存文件的全路径
21	close( )	方法	Void	关闭文件
22	quit( )	方法	Void	退出程序

### 3 WordWriter 组件的应用

#### 3.1 WordWriter 组件部署

要在程序中使用 WordWriter 组件,要经过以下步骤:

1) 下载 JACOB 项目包,如 jacob1.2.zip,具体参见 <http://danadler.com/jacob/>.

2) 将下载的文件解压缩,把其中的 jacob.jar 文件复制到本机的 JDK 目录的 lib 文件夹下,并在环境变量中指明路径.另外还要把其中的 jacob.dll 文件复制到本机 Windows 目录中的 System32 文件夹下.

3) 在 Java 程序中引入 WordWriter 组件类.

4) 在 Java 程序中声明 WordWriter 组件的对象,调用使用.

#### 3.2 WordWriter 组件应用步骤

笔者参与的项目中大量使用了利用 JACOB 组件生成固定格式的 Word 文件.使用 JACOB 组件生成 6 种格式的 Word 文件大约要使用 2 000 行的代码,而用 WordWriter 组件来实现大约只需要 700 ~ 750 行的代码.归纳一下,使用 WordWriter 组件的步骤大致如下:1) 初始化 WordWriter 组件的实例;2) 创建 Word 文档;3) 设置段落和字体的格式;4) 填入相应内容;5) 保存和关闭文档;6) 释放 WordWriter 对象实例.

#### 3.3 WordWriter 组件应用实例

本文选取某办公系统中的需要生成固定格式的 Word 文档为例:系统要求生成的 Word 文档有多个,有通知文件、报表文件、申请书文件等.选取通知文件来用 WordWriter 组件生成.需求要求:“通知”文字居中显示、黑体不加粗、三号字体,通知内容为“仿宋”不加粗、小四,签发单位为右对齐、“仿宋”不加粗、小四,时间为右对齐、不加粗、小四:

```
WordWriter objWordWriter = new WordWriter();
objWordWriter.createNewWordDocument();
objWordWriter.createNewParagraph(1);
objWordWriter.setFontStyle("黑体", false, false, false, "0,0,0,0", 16);
objWordWriter.setAlignment();
objWordWriter.insertContents("通知");
objWordWriter.createNewParagraph();
objWordWriter.setFontStyle("仿宋", false, false, false, "0,0,0,0", 12);
```

```
objWordWriter.setAlignment(0);
objWordWriter.insertContents("通知内容");
objWordWriter.createNewParagraph();
objWordWriter.setAlignment(2);
objWordWriter.insertContents("签发单位");
objWordWriter.createNewParagraph();
objWordWriter.setAlignment(2);
objWordWriter.insertContents("签发时间");
objWordWriter.saveAs(path);
objWordWriter.close();
objWordWriter.quit();
```

### 4 结语

现在 Java 开发软件中对 Word 编程的要求越来越多,但 Java 在这方面提供的方法却不是丰富.即使在 JACOB 出现以后,这样的问题依然存在.本文在 JACOB 的基础上,对实际项目中需要的功能进行了总结抽象,结合 CBSD 的开发思想设计并实现了基于 Java 的 WordWriter 组件.具备松耦合特性的 WordWriter 组件可以方便地在 Java 代码中生成各种格式的 Word 文件,大大缩短了软件开发的周期,降低了后期的软件维护成本,同时使得代码变得清晰简洁.本组件已在多个项目中获得应用,具有较强的通用性.

#### 参考文献:

- [1] Dan Adler. The JACOB Project [EB/OL]. (2004 - 10 - 17) [2011 - 03 - 09]. <http://danadler.com/jacob/index.html>.
- [2] Roger S. Software Engineering A Practitioner's Approach [M]. 北京:清华大学出版社,2003:722.
- [3] Cao Min, Wang Yanyan, Wu Gengfeng. A reuse approach of description for component based distributed software architecture [J]. Comp Eng and Appl, 2005, 15: 94.
- [4] Shinkawa Y, Masao J Matsumoto. Software Reuse: Advances in Software Reusability [M]. Heidelberg: Springer, 2000: 9 - 37.
- [5] 李瑞,李永刚. Java 中基于 JACOB 的 COM 组件调用研究 [J]. 微计算机信息, 2007, 24(5): 168.
- [6] 张金波,孙海翠. COM 技术及其程序设计 [J]. 河海大学常州分校学报, 2004, 18(2): 34.
- [7] 杨晓峰,仇璐. 通过 JACOB 实现 Java 与 COM 组件的互操作 [EB/OL]. (2009 - 12 - 08) [2011 - 03 - 09]. <http://newitfarmer.com/bbs/read.php?tid-18863.html>.