Word 服务器的接口解析与文档属性提取

汤克明[1,2],陈 崚[3]
TANG Ke-ming[1,2],CHEN Ling[3]

1.南京航空航天大学 信息科学与技术学院,南京 210016
2.盐城师范学院 计算机系,江苏 盐城 224002
3.扬州大学 计算机科学与工程系,江苏 扬州 225009
1.College of Information Science and Technology,Nanjing University of Aeronautics and Astronautics,Nanjing 210016,China
2.Department of Computer,Yancheng Teachers University,Yancheng,Jiangsu 224002,China
3.Department of Computer Science and Engineering,Yangzhou University,Yangzhou,Jiangsu 225009,China
E-mail:tkmchina@126.com

TANG Ke-ming,CHEN Ling.Analysis of interface and extracting of document attribute for Word server.Computer Engineering and Applications,2008,44(28):79-82.

Abstract: This paper discourses on the necessity of analysis on object library of Office system.For example with word software, word server is summarized,and the interfaces and corresponding implementation classes of seven components in Word server are gived,at the same time,the hiberarchy of the main interface objects is explained.Lastly,the methods used to extract document attributes of Word are presented.And by means of implementation of word-document analysis tool,the practice shows that the above methods are true and feasible.

Key words: word server;analysis of interface;extracting of document attribute

摘 要:论述了对 Office 对象库分析的必要性;以 Word 软件为例,对 Word 服务器进行了概述,给出了构建 Word 服务器的 7 个组件接口以及对应的实现类,并对主要接口对象之间的层次关系进行了说明;介绍了 Word 文档属性的提取方法,通过 Word 文档分析工具的实现证明所给方法是正确并且可行的。

关键词: Word 服务器;接口解析;文档属性提取
DOI:10.3778/j.issn.1002-8331.2008.28.028 **文章编号:**1002-8331(2008)28-0079-04 **文献标识码:**A **中图分类号:**TP317

1 引言

随着社会信息化进程的加快,大量信息在给人们带来方便的同时也带来了一些问题,比如:信息量过大,超过了人们掌握与消化的能力;信息组织形式的不一致性导致难以对信息进行有效统一处理等等。至于此,人们研究数据挖掘(Data Mining)理论与技术, 从海量而复杂的数据中提取出有价值的知识,以进一步提高信息的利用率; 研究企业应用集成(Enterprise Application Integration)技术,对企业业务过程进行有效重构和柔性化管理,从宏观上控制多个应用系统的关系,对诸多应用系统组成的整体进行优化。

微软 Office 套件因满足办公需求,已经成为十分流行的应用软件。据统计有超过数千万的用户经常使用 Office 软件来完成他们的工作。Office 软件所处理的数据已成为数据挖掘的重要对象之一,基于 Office 的自动化编程也成为企业应用集成所考虑的重要内容之一。每一个 Office 应用程序具有可编程性,支持同其他程序整合的功能,便于人们开发高集成的软件并提供完美的数据与信息共享功能。开发集成的 Office 解决方案主要依靠两类技术:Microsoft Visual Basic for Application(VBA)和组件对象模型(COM)软件构建。由于 COM 具有语言的无关性、进程的透明性、可重用性以及安全性,因而得到更多的关注。基于 COM 的 Office 应用系统的自动控制与集成的工作基础是理解自动化技术,其核心是掌握 Office 套件中的各个应用服务器的接口、对象以及对象模型。作为一个编程环境可以通过访问一个对象库(Object Library)来决定一个对象的特征,比如对象所支持的接口和接口的名称以及接口层次。有了这些信息,编程语言就可以用来同提供的接口进行工作。由于微软公司并没有提供对象库中有关接口与对象模型的说明文档,对类型库进行解析是一项有很意义的工作。本文以 Word2000 服务器为例,借助于 Delphi 平台中的 Word2000.pas 对象库对 Word 服务器接口进行解析,针对 Word 文档分析工作给出文档属性

基金项目:国家自然科学基金(the National Natural Science Foundation of China under Grant No.60673060);江苏省自然科学基金(the Natural Science Foundation of Jiangsu Province of China under Grant No.BK2005047)。

作者简介:汤克明(1965-),男,博士研究生,副教授,主要研究领域为对等计算,人工智能;陈崚(1951-),男,教授,博士生导师,主要研究领域为并行与分布计算,人工智能。

收稿日期:2008-01-23 **修回日期:**2008-04-02

的提取方法。

2 Word 服务器的一般理解

Word 服务器是通过可编程的组件对象模型实现的，如 Application 对象、Document 对象以及 Paragraph 对象等，每一个对象完成 Word 服务器的部分功能。Word 的最基本工作对象是文档，并且任何工作都是文档的一部分。由于字符构成单词，单词构成句子，句子构成段落，最终形成文档，因此，每个 Document 对象都有一个 Characters 集合、一个 Words 集合、一个 Sentences 集合和一个 Paragraphs 集合。并且，每个文档还有一个或多个 Sections 集合，每个节还有一个 HeaderFooters 集合。所谓集合，就是一组相似对象的组合，它的重要属性是可以对其进行整体操作。

Word 服务器的应用级对象是 Application 对象自身与它的属性、方法以及 Options 选项，还有一些 Word 内置对话框。Options 对象包含了可以实现自定义 Word 的外观和行为方式的功能设置。应用级对象能作用于多个文档，或者可以访问或操作当前独立的、活动的一个 Document 对象。对文档内容进行操作，首先应确定文档的一个区域，才能进行操作。Rang 对象通过定义首字符和尾字符位置来确定文档中的一个相邻区域。Selection 对象表示 Word 文档中当前选择的文本。Bookmark 对象与 Selection 对象或 Rang 对象相同，也表示文档中的一段相邻区域。

Word 服务器通过接口方法对外提供服务，服务的功能由实现接口方法的若干组件对象完成。

3 Word 服务器的接口解析

通过分析 Word2000 对象库，可以解析出 Word2000 服务器所包含的接口以及实现接口的对象之间的层次关系。Word 对象类型库标识为 LIBID_Word，它的 GUID 为：

'00020905-0000-0000-C000-000000000046'

在 Word2000 对象库中，定义了 198 个非派遣接口，例如：Word 应用服务器接口标识为 IID_Application，它的 GUID 为：

'00020970-0000-0000-C000-000000000046'

对应的接口名为"_Application"；定义了 4 个派遣接口，例如：Word 应用服务器事件接口标识为 DIID_ApplicationEvents，它的 GUID 为：

'000209F7-0000-0000-C000-000000000046'

对应的接口名为"_ApplicationEvents"；定义了 7 个实现接口方法的组件类，例如：Word 文档组件类标识为 CLASS_WordDocument，它的 GUID 为：

'00020906-0000-0000-C000-000000000046'

对应的组件类名为"WordApplication"。

其中，由 7 个 COM 组件所实现的 7 个 Word 服务器接口的对应关系如表 1 所示，创建 7 个接口的 7 个类工厂对应关系如表 2 所示。

Word 服务器的 198 个接口对象有一个层次关系，根节点是 _Application 接口对象，代表惟一的 Word 应用程序实例。通过查阅对象库可以了解到，_Application 接口具有 Language、Windows、Documents 以及 Section 等多个接口属性，从而 _Application 接口对象下面有 Documents 接口对象。Documents 集合接口对象可以通过它的 Item 方法得到 _Document 接口对象。_Document 接口具有：Paragraphs、Sentences 以及 Words 等多个接口属性，从而 _Document 接口对象又可以访问 Paragraphs、Sentences 以及 Words 等对象。通过创建顶层对象即可以向下逐层访问下层对象，当然下层对象也可以通过 Parent 方法得到上层对象。由于篇幅限制，另文详细介绍接口对象之间的关系。在此，通过下文对文档属性提取的描述，也可以了解 Word 服务器的主要接口对象之间的关系。

表 1 服务器接口与对应的实现类

服务器接口	接口的实现类
_Application	WordApplication
_Document	WordDocument
_Font	WordFont
_Global	Global
_LetterContent	WordLetterContent
_OLEControl	WordOLEControl
_ParagraphFormat	WordParagraphFormat

表 2 服务器接口与对应的类工厂

服务器接口	类工厂
_Application	CoWordApplication
_Document	CoWordDocument
_Font	CoWordFont
_Global	CoGlobal
_LetterContent	CoWordLetterContent
_OLEControl	CoWordOLEControl
_ParagraphFormat	CoWordParagraphFormat

可以通过 CoWordApplication 类工厂来创建 Word 应用程序实例，即：

t_wordapp:=cowordapplication.Create;

通过 t_docs:=t_wordapp.Documents 创建文档集合对象；再通过语句：t_doc1:=t_docs.Open(t_filename1,ConfirmConversions,ReadOnly,AddToRecentFiles,PasswordDocument,PasswordTemplate,Revert,WritePasswordDocument,WritePasswordTemplate,Format,Encoding,doc_Visible);来创建一个文档对象 t_doc1。

4 Word 文档属性的提取

4.1 提取文档的一般属性

(1)提取文档名称：t_doc1.Name；(2)提取文档路径：t_doc1.Path；(3)提取文档类型：t_doc1.Type_；(4)提取段落数目：t_doc1.Paragraphs.Count；(5)提取节的数目：t_doc1.Sections.Count；(6)提取注释数目：t_doc1.Comments.Count；(7)提取脚注数目：t_doc1.Footnotes.Count；(8)提取尾注数目：t_doc1.Endnotes.Count；(9)提取书签数目：t_doc1.Bookmarks.Count；(10)提取表格数目：t_doc1.Tables.Count；(11)提取图片数目：t_doc1.Shapes.Count。

4.2 提取页面设置属性

提取条件是得到节对象，方法是：

t_section:= t_doc1.Sections.Item(t_section_no).

4.2.1 关于页边距

(1)提取上边距：t_section.PageSetup.TopMargin；(2)提取下边距：t_section.PageSetup.BottomMargin；(3)提取左边距：t_section.PageSetup.LeftMargin；(4)提取右边距：t_section.PageSetup.RightMargin；(5) 提取装订线边距：t_section.PageSetup.Gutter；(6)提取装订线位置：t_section.PageSetup.GutterPos；(7)提取页面方向：t_section.PageSetup.Orientation。

4.2.2　关于纸张

(1)纸张大小:t_section.PageSetup.PaperSize;(2)页面宽度:t_section.PageSetup.PageWidth;(3)页面高度:t_section.PageSetup.PageHeight;(4)首页纸张来源:t_section.PageSetup.FirstPageTray;(5)其它纸张来源:t_section.PageSetup.OtherPagesTray。

4.2.3　关于版式

(1) 页面版式按节进行定义, 节的开始位置:t_section.PageSetup.SectionStart;(2) 是否取消尾注:t_section.PageSetup.SuppressEndnotes;(3)页眉距边界:t_section.PageSetup.HeaderDistance;(4)页脚距边界:t_section.PageSetup.FooterDistance;(5) 是否选择页眉页脚奇偶页不同方式:t_section.PageSetup.OddAndEvenPagesHeaderFooter;(6)是否选择页眉页脚首页不同方式:t_section.PageSetup.DifferentFirstPageHeaderFooter;(7)垂直对齐:t_section.PageSetup.VerticalAlignment。

4.2.4　关于文档网格

(1)每页行数:t_section.PageSetup.LinesPage;(2)每行字数:t_section.PageSetup.CharsLine。

4.2.5　关于行编号

首先得到当前节的行编号对象,方法是:

t-line:=t_section.PageSetup.LineNumbering;

(1)是否选择添加行号:t-line.Active;(2)起始编号:t-line.StartingNumber;(3)行号与正文的距离:t-line.DistanceFromText;(4)行号间隔:t-line.CountBy;(5)编号方式:t-line.RestartMode。

4.3　提取字体属性

提取条件是得到区域对象与字体对象,方法是:

t_range:=t_doc1.Range(t_start,t_end);

t_wordfont:=t_range.get_font。

4.3.1　关于字体

(1)提取区间文本:t_range.Text;(2)提取中文字体:t_wordfont.Name;(3)提取西文字体:t_wordfont.NameAscii;(4)提取字号:t_wordfont.Size;(5)是否加粗:t_wordfont.Bold;(6)是否倾斜:t_wordfont.Italic;(7)是否隐藏:t_wordfont.Hidden;(8)字体颜色:t_wordfont.ColorIndex;(9)下划线线型:t_wordfont.Underline;(10)下划线颜色:t_wordfont.UnderlineColor;(11)是否加着重号:t_wordfont.EmphasisMark。

4.3.2　关于效果

(1)是否加单删除线:t_wordfont.StrikeThrough;(2)是否加双删除线:t_wordfont.DoubleStrikeThrough;(3)是否上标:t_wordfont.Superscript;(4)是否下标:t_wordfont.Subscript;(5)是否阴影:t_wordfont.Shadow;(6)是否空心:t_wordfont.Outline;(7)是否阳文:t_wordfont.Emboss;(8)是否阴文:t_wordfont.Engrave;(9)是否小型大写字母:t_wordfont.SmallCaps;(10)是否全部大写字母:t_wordfont.AllCaps;(11)是否加动态效果:t_wordfont.Animation;(12)是否隐藏文字:t_wordfont.Hidden。

4.3.3　关于字符间距

(1)字符缩放比例:t_wordfont.Scaling;(2)字符间距:t_wordfont.Kerning;(3)字符位置:t_wordfont.Position。

4.4　提取段落属性

提取条件是得到段落对象,方法是:

t_pgh:=t_doc1.Paragraphs.Item(t_pgh_no)。

4.4.1　关于段落常规

(1)段落文本:t_pgh.Range.Text;(2)对齐方式:t_pgh.Alignment;(3)大纲级别:t_pgh.Outlinelevel。

4.4.2　关于段落缩进

(1)段落左缩进:t_pgh.LeftIndent;(2)段落右缩进:t_pgh.RightIndent;(3)段落首行缩进:t_pgh.FirstLineIndent;(4)是否选择自动调整右缩进:t_pgh.AutoAdjustRightIndent。

4.4.3　关于间距

(1)段落行距:t_pgh.LineSpacing;(2)段落前空:t_pgh.SpaceBefore;(3)段落后空:t_pgh.SpaceAfter。

4.4.4　关于换行与分页

(1)是否孤行控制:t_pgh.WidowControl;(2)是否与下段同页:t_pgh.KeepWithNex;(3)是否段中不分页:t_pgh.KeepTogether;(4)是否段前分页:t_pgh.PageBreakBefore;(5)是否取消行号:t_pgh.NoLineNumber。

4.4.5　关于中文版式

(1)是否按中文习惯控制首尾字符:t_pgh.FarEastLineBreakControl;(2)是否允许西文在单词中间换行:t_pgh.WordWrap;(3)是否允许标点溢出边界:t_pgh.HangingPunctuation;(4)是否允许行首标点压缩:t_pgh.HalfWidthPunctuationOnTopOfLine;(5)是否自动调整中文与西文间距:t_pgh.AddSpaceBetweenFarEastAndAlpha;(6)是否自动调整中文与数字间距:t_pgh.AddSpaceBetweenFarEastAndDigit;(7)文字对齐方式:t_pgh.BaseLineAlignment。

4.4.6　关于首字下沉

(1)首字符位置:t_pgh.DropCap.Position;(2)首字符字体名称:t_pgh.DropCap.FontName;(3)首字符下沉行数:t_pgh.DropCap.LinesToDrop;(4)首字符距正文距离:t_pgh.DropCap.DistanceFromText。

4.5　提取边框属性

针对页面边框, 提取条件是得到节对象与边框对象,方法是:

t_sections:=t_doc1.Sections;

t_section:=t_sections.Item(t_section_no);

t_borders:=t_section.Borders。

针对段落边框, 提取条件是得到段落对象与边框对象,方法是:

t_pgh:= t_doc1.Paragraphs.Item(t_pgh_no);

t_borders:=t_pgh.Range.Borders。

针对文字边框, 提取条件是得到区域对象与边框对象,方法是:

t_range:=t_doc1.Range(t_start,t_end);

t_borders:= t_range.Borders。

(1)是否加边框:t_borders.Enable;(2)是否阴影边框:t_borders.Shadow;(3)边框上边距:t_borders.DistanceFromTop;(4)边框左边距:t_borders.DistanceFromLeft;(5)边框右边距:t_borders.DistanceFromRight;(6)边框下边距:t_borders.DistanceFromBottom;(7)边框内部线型:t_borders.InsideLineStyle;(8)边框外部线型:t_borders.OutsideLineStyle;(9)边框内部线宽:t_borders.InsideLineWidth;(10)边框外部线宽:t_borders.OutsideLine Width;(11)边框内线颜色:t_borders.InsideColorIndex;

(12)边框外线颜色:t_borders.OutsideColorIndex。

4.6 提取底纹属性

针对段落底纹，提取条件是得到段落对象与底纹对象,方法是:

t_pgh:=t_doc1.Paragraphs.Item(t_pgh_no);

t_shading:=t_pgh.Range.Shading。

针对文字底纹，提取条件是得到区域对象与底纹对象,方法是:

t_range:=t_doc1.Range(t_start,t_end);

t_shading:=t_range.get_shading。

(1)底纹前景色:t_shading.ForegroundPatternColorIndex;(2)底纹背景色:t_shading.BackgroundPatternColorIndex;(3)底纹纹理:t_shading.Texture。

4.7 提取页眉页脚属性

提取条件是得到节、页眉、页脚以及字体等对象,方法是:

t_sections:=t_doc1.Sections;

t_section:=t_sections.Item(t_section_no);

t_header:=t_section.Headers.Item(i);

t_footer:=t_section.Footers.Item(i);

t_headPgh:=t_header.Range.Paragraphs.Item(i);

t_header_wordfont:=t_header.Range.get_font;

t_footPgh:=t_footer.Range.Paragraphs.Item(i);

t_footer_wordfont:=t_footer.Range.get_font。

4.7.1 关于页眉

(1)页眉是否与前节相同:t_header.LinkToPrevious;(2)页眉文本:t_header.Range.Text;(3)页眉文本字体:t_header.Range.Font.Name;(4)页眉文本字号:t_wordfont.Size;(5)页眉文本对齐方式: t_headPgh.Alignment;(6)页眉文本颜色:t_header_wordfont.ColorIndex;(7)页眉文本是否加粗:t_header_wordfont.Bold;(8)页眉文本是否倾斜:t_header_wordfont.Italic。

4.7.2 关于页脚

(1)页脚是否与前节相同:t_footer.LinkToPrevious;(2)页脚文本:t_footer.Range.Text;(3)页脚文本字体:t_footer.Range.Font.Name;(4)页脚文本字号:t_footer_wordfont.Size;(5)页脚文本对齐方式:t_footPgh.Alignment;(6)页脚文本颜色:t_footer_wordfont.ColorIndex;(7)页脚文本是否加粗:t_footer_wordfont.Bold;(8)页脚文本是否倾斜:t_footer_wordfont.Italic。

4.8 提取分栏属性

提取条件是得到节与文本栏对象,方法是:

t_sections:=t_doc1.Sections;

t_section:=t_sections.Item(t_section_no);

t_columns:=t_section.PageSetup.TextColumns;

t_column:=t_columns.Item(i)。

(1)文本栏数:t_columns.Count;(2)是否选择栏宽相等:t_columns.EvenlySpaced;(3)是否选择分隔线:t_columns.LineBetween;(4)当前栏宽度:t_columns.Width;(5)当前栏后空:t_column.SpaceAfter。

4.9 提取图片属性

提取条件是得到段落与图片对象,方法是:

t_pgh:=t_doc1.Paragraphs.Item(t_pgh_no);

t_shape:=t_pgh.Range.ShapeRange.Item(i)。

(1)高度:t_shape.Height;(2)宽度:t_shape.Width;(3)水平翻转:t_shape.HorizontalFlip;(4)垂直翻转:t_shape.VerticalFlip;(5)左裁剪:t_shape.PictureFormat.CropLeft;(6)右裁剪:t_shape.PictureFormat.CropRight;(7)上裁剪:t_shape.PictureFormat.CropTop;(8)下裁剪:t_shape.PictureFormat.CropBottom;(9)颜色:t_shape.PictureFormat.ColorType;(10)亮度:t_shape.PictureFormat.Brightness;(11)对比度:t_shape.PictureFormat.Contrast;(12)环绕方式:t_pgh.Range.ShapeRange.WrapFormat.Type_;(13)图片上方到正文的距离:t_pgh.Range.ShapeRange.WrapFormat.DistanceTop;(14)图片下方到正文的距离:t_pgh.Range.ShapeRange.WrapFormat.DistanceBottom;(15)图片左方到正文的距离:t_pgh.Range.ShapeRange.WrapFormat.Distance-Left;(16)图片右方到正文的距离:t_pgh.Range.ShapeRange.WrapFormat.DistanceRight。

5 结束语

本文对 Word 服务器的接口进行了解析，并给出了 Word 文档常规属性的提供方法。尽管由于篇幅限制,未能详细介绍,特别是接口对象之间的层次关系,但是读者完全可以根据所述内容理解其本质内容。作者根据所述知识,在 Delphi7.0 平台下开发了一个实用的 Word 文档分析工具，其界面如图 1 所示。实践证明基于文章所述内容,可以方便地进行 Word 自动化编程,也可把所给方法用于文档聚类研究中涉及关键词频度与深度等特征测量方面。

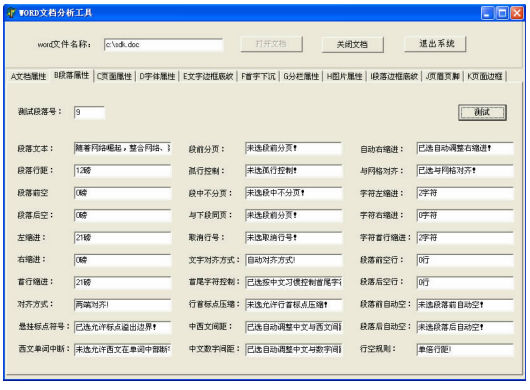

图 1 Word 文档分析工具界面

参考文献:

[1] Brockschmidt K.How OLE and COM solve the problems of component software design[J].Microsoft Systems,1996,15(5):63-82.

[2] Sullivan K J,Marchukov M,Socha J.Analysis of a conflict between aggregation and interface negotiation in Microsoft's component object model[J].IEEE Transactions on Software Engineering,1999,25(4):584-599.

[3] 陈旭,杨彬,刘怀.Delphi COM 深入编程[M].北京:机械工业出版社,2000.

[4] Spott M,Nauck D.Towards the automation of intelligent data analysis[J].Applied Soft Computing,2006,6(8):348-356.