一种解析和处理 PDF 格式文档的解决方案

李福全，杨俊，满成圆
北京邮电大学计算机科学与技术学院，北京（100876）
E-mail: lfqhaoyun@gmail.com

摘　要：本文提出了一个将 PDF 文件转换成 XML 格式的解决方案，以使程序设计人员能够方便地处理 PDF 文件，将所有 PDF 文件转换成 XML 格式，并可进而将 XML 格式转换为 HTML、WML 等格式，来方便 PDF 文件在传统互联网和移动互联网上的传播，简化对 PDF 格式文档的信息收取，自动摘要，全文检索等操作。

关键词：PDF，ppXML 格式，easyPDF 系统，PDF 字体，CMap
中图分类号: TP391　　**文献标识码: A**

1. 引言

PDF(Portable Document Format)[1]由于其排版不受操作系统的影响，不容易受病毒感染等特点受到了人们的青睐，是政府机构、企业特别是学术界最常用的文档格式之一。PDF 采用二进制流与纯文字混合的编码模式，并且没有采用 Unicode 等标准字符编码方式，其字符编码采用 Adobe 公司内建的编码表（CMap），这使得对 PDF 的处理更加困难。

目前的 PDF 文档项目，分为商业用与开源两类。在商业用的工具当中，Adobe 公司的工具是最为成熟的。除了 Adobe 公司之外，还有许多商业公司开发出各式各样的 PDF 工具，例如 SolidConvertPDF 等。在开源领域，iText[3]、Multivalent[4]、PDFBox[5]、PJX[6]、XPDF[7]等工具提供一些 PDF 工具可以将文字、图片等对象从 PDF 文档中抽出来。

但除了 Adobe SDK 之外，其他商业工具经常无法将所有的 PDF 文档正确的转换成纯文本文件；开源的工具都无法正确的处理中文、日文、韩文等东亚字体，转换后会得到一堆乱码。另外目前还没有可以将 PDF 中的字号、位置等信息抽取出来的 PDF 工具。由此可见，现有的工具无法满足对 PDF 文档操作的需要。

本文提出了一种使用 Java 语言完成的将 PDF 转换成 ppXML 格式的解决方案——easyPDF 文档转换系统。该方案可以将文档格式转换为 ppXML 格式。ppXML 是一种 XML 文档格式，包含源文档的内容和格式信息，是一种标准化的格式，其设计目标是让对文档的操作更容易，而不需要理解各种文档的详细结构。ppXML 作为一种中间格式，来解析 PDF 文档格式的内容，使得 PDF 文档的信息可以得到重复利用，本系统还提供将 ppXML 转换为 HTML 与 WML 等格式的功能，使得 PDF 格式文档不再受限于特定的阅读器，而能够在互联网终端和移动设备上进行阅读。

本文第二部分介绍了 PDF 格式的结构和 PDF 解析器的建立,第三部分给出了 easyPDF 方案的系统架构，然后重点讲解了 easyPDF 对 PDF 字体的处理过程,最后本文对 easyPDF 进行了客观的系统分析和总结。

2. PDF 文档格式原理及解析器的构建

PDF 文档格式转换的关键之一是要构建 PDF 文档解析器。根据 PDF 的格式参考[1]可知，PDF 格式里面包含各种内部对象，这些对象来描述 PDF 文档的各种信息，如文字、图片、表格等。内部对象的组织关系由 PDF 的文件结构和文档结构来定义。

2.1 PDF 格式的结构简介

PDF 的文件结构（即物理结构）包括四个部分：文件头、文件体、交叉引用表和文件尾。文件头(Header)指明了该文件所遵从 PDF 规范的版本号，它出现在 PDF 文件的第一行；文件体(Body)由一系列的 PDF 间接对象组成，这些间接对象构成了 PDF 文件的具体内容；交叉引用表(Cross- reference Table)则是为了能对间接对象进行随机存取而设立的一个间接对象地址索引表；文件尾(Trailer)声明了交叉引用表的地址，指明文件体的根对象 (Catalog)，还保存了加密的安全信息。根据文件尾提供的信息，PDF 的应用程序可以控制整个 PDF 文件。

PDF 的文档结构（即逻辑结构），反映了文件体中间接对象间的等级层次关系。PDF 的文档结构是一种树型结构。树的根节点就是 PDF 文件的根对象。根节点下有四个子树：页面树(Pages Tree)、书签树(Outline Tree)、线索树(Article Threads)、名字树(Named Destination)。页面树节点是文档页面树的根节点，所有页面对象都在树的叶子节点，树中的子节点将继承父节点的各属性值作为相应属性的缺省值；书签树中则按树形层次等级关系将书签（Book mark）组织起来，书签建立了书签名与一个具体页面上的位置的关联，它使得用户可以按书签名字来访问文档的内容；线索树是将文章线索以及线索下的各文章块（Article Bead），按照树型的结构组织起来进行管理；名字树，它是建立了一种字符串（即名字）和页面区域的对应关系，树中的各叶子节点保存着字符串及其相应的页面区域，而非叶子节点则只是一种索引。

2.2 构造 PDF 解析器

PDF 的核心是页面描述语言(Page Description Language)[1]，这是一个以树状结构为主的语法，树中节点采用类似对象导向的设计。首先初始化文件结构找到文件尾，从文件尾字典中能得到文件体的根对象偏移地址和交叉引用表的偏移地址。

Catalog 是整棵树的根节点，对 PDF 文档剖析，找到根节点，然后就可以利用 Pages 信息找到所有 Page 对象，从而得到 Contents Stream、Resources Dictionary 等内容，从而完成对整个文档完成解析。其中，Contents 代表页面中的内容，其中的 Stream 记载了每个页面中的文字、图片等信息，以及这些对象的坐标转换指令，这些指令包含坐标定位、对象旋转、放大缩小等等与排版有关的转换；Resources 该项列出了内容流中用到的资源项，并建立了一个资源名字与资源对象本身的映射表。这样就完成了 PDF 解析器的构建。

3. easyPDF 系统实现

根据前面介绍的相关原理以及 PDF 解析器的构造，本为提出了一个解析和转换 PDF 格式文档的解决方案 easyPDF。.图 1 为 easyPDF 系统转换流程图。

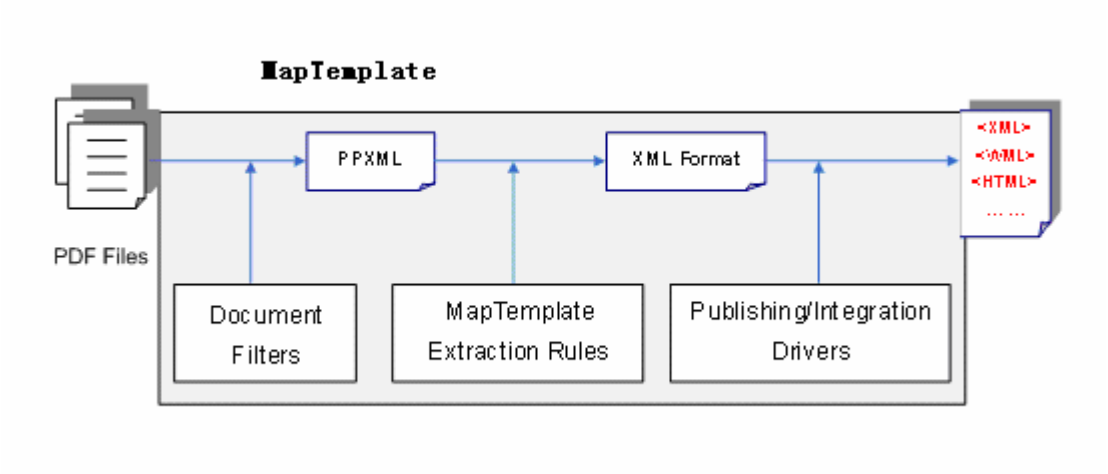

图 1　easyPDF 系统转换流程图

easyPDF 转换系统的内部结构如图 2 所示。PDF 文档编辑器负责提供创建、修改和保存文档的编辑环境。PDF 文档解析器负责 PDF 文档的逻辑结构和数据结构的分析，根据解析的结果构造文档树。XML 解析器负责 XML 文档的解析和 DOM 树[2]的构造。在 PDF 文档转换器中，文档转换过程首先调用源文档树，然后根据转换规则将源文档树转换为目标文档树，最后将目标文档树写成目标文档并送到相应的编辑环境中或在浏览器中显示。这三个部分是相对独立的，可以更为深层次的将文档转换过程细化，具有可扩展性，执行过程清晰、简练。其中文档转换的核心是文档解析器、文档转换器和文档树，文档树包括 easyPDF 文档树和 DOM 树。

如图 2 所示，解析器建立后，系统将对 PDF 文档逐页进行虚拟渲染，得到各个页面以后，通过 Page 的页面指令我们可以得到页面的 Contents Stream，Resources 字典等信息，并按照 Contents Strean 中的操作符按照相应的转换规则建立 Dom Tree。

PDF 对字体的处理相当复杂，因此，easyPDF 对 PDF 字体处理部分也就成为整个模块的关键部分和技术，下面详细讲解一下 PDF 字体方面的解析方法。

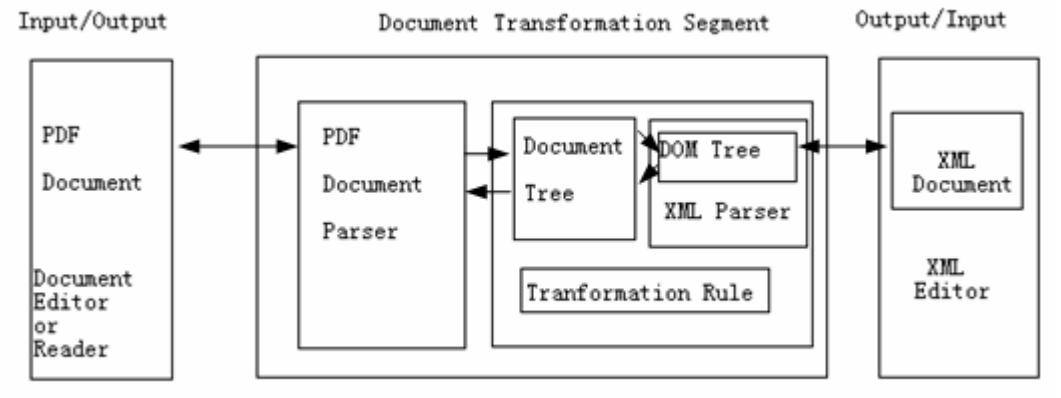

图 2　easyPDF 转换系统内部结构图

3.1 CMap

CMap 是一种字符映射表。CMap 记载了各类字码与 PDF 内码的对应方式，例如：GB-EUC-H 记载了 GB2312 中文码对应到 PDF 内码的方式。CMap 也指出了写作模式，垂直

或者是水平，写作模式指定了当显示字形时的渲染方式。

PDF 采用的编码集种类非常之多。例如 PDF 的简体中文编码，就有 UniGB-UCS2-H，UniGB-UCS2-V，GB-EUC-H，GB-EUC-V， GBpc-EUC-H，GBpc-EUC-V，GBK-EUC-H，GBK-EUC-V，GBK2K-H，及 GBK2K-V 等字符集，而且还可以将 CMap 直接内嵌在 PDF 文档中成为一个对象，而不需要遵照任何一种编码规则，这些复杂的编码方法增加了 PDF 译码的困难度。

3.2 PDF 字体的一般解析流程

PDF 字体解析一般分为两步来进行。

首先根据 TextState 系列指令获得字体的文本状态属性，如字符间距、单词间距、字体宽高比、字体渲染模式、根据 **Tf** 操作符的 Font Name 条目中得到属性名并从 Resource 字典中得到与 Font Name 相映的 Font 字典及字体字号等；从文本位置系列指令得到字体的坐标信息及其它一些位置信息；从文本显示系列指令得到字体的字节码及显示的一些其它控制信息。

然后根据得到的字节码流和 Font 字典信息对字节码进行解码。解码字节码过程：根据 Font 字典中的 Encoding 信息将文本的字节码流转换为 PDF 的标准内码 CID (Character Identity)，最后再利用 Font 字典中的 ToUnicode 中的包含的 CMap 信息将 CID 转换为 Unicode 编码，然后将字节码流解码，如此即可抽取出 PDF 文档中的文字，再利用 Font 字典中的 Font Descriptors 条目中的信息设置字体的其它一些信息如是否斜体、是否加粗等，利用 BaseFont 条目得到字体的名称信息（如宋体，楷体等），设定文本的位置，并用渲染器写入 ppXML 文件里面。

但在实际中，现存的 PDF 文件的字体信息并不是每种字体都包含 Enconding 和 ToUnicode 属性，更复杂的情况是 Stream 的内容通常是压缩过的，在解码 Stream 之前，必须先检查是否被压缩过。另一个难以处理的问题是，PDF 中的对象并非以树状的层次结构直接储存，而是以类似超链接的方式所形的树状结构，如何连结并没有限制，这一问题可以通过交叉引用表的信息来加以控制。

3.3 各种字体的解码的异同点

PDF 的字体大致分为 4 种字体 TrueType, Type1, Type3, 以及 Type0 字体，前三种字体又称简单字体，Type0 字体又称合成字体[1]。这四种字体除了 Type1 字体和 TrueType 字体的字典项目属性基本相同外，其余的几种字体各有其不同的项目属性，必然这些字体的各自私有的项目属性让不同的字体的解码方式各有差异。Type0 字体是最难处理的一种字体，它的字形由一种类似字体的 CIDFont 对象得到。Type0 字体是一种为高效方便定义多字节编码以及包含大量字形的字体而定义的，这就使定义中日韩等字体变得可行而有效，但是与此方便形成鲜明对比的就是其解码过程也就变得相当复杂。

下面来详细说明一下这四种字体解码的相同点和不同点。

相同点：当字体字典包含 ToUnicode 项目属性时，直接按照 ToUnicode 项目属性的流对象将 CMap 构建出来，利用构建的 CMap 进行解码。

不同点：当字体字典不包含 ToUnicode 项目属性的时候，各种字体解码的方式各不相同:

1、简单字体（Type1，Type3，TrueType）解码方法：

A、 Endcoding 为预定义的的字符集的 Name，并按此字符集解码

B、 Encoding 为 Dictionary 类型，则应用其内包含的 Difference 数组和 BaseFontName 进行解码

C、 当 Encoding 属性为空时，采用默认的编码集解码

2、合成字体（Type0）解码方法：

A、Encoding 为 Predefined CMap，利用这些 CMap 进行编码

B、Encoding 为 Generic（即 Name 为 Identity-H 或 Identity-V）则根据其 CID FontInfo 选择一种 CMap 对其进行解码

3、每种字体传入的字节码有 Literal String 和 Hex String 之分[1]，这两种类型的字节码的解码区别主要在于解码时解码字节长度的不同。

4、 Type3 字体的 Font 字典包含字形显示的信息，需单独予以处理。

4. 系统分析

在现有 PDF 转换工具当中，有些可以将 PDF 中的文字抽取出来，经过测试 (在 MS Windows XP 上)，这些转换工具的在处理各种字体的文件上，测试结果如下：

表 1 测试结果

软件名称	英文转纯文字	东亚字体转纯文字	输出字号 与坐标信息
Solid Converter PDF	正常	可以、但部分文档转换失败、没有任何输出	无
iText	无	无	无
PDF Box	正常	东亚字体支持很差	无
Multivalent	正常	东亚字体全部变乱码	无
PJX	正常	东亚字体支持很差	无
easyPDF	正常	正常	有

由上表可知，easyPDF 的优势在于东亚（CJK）字体的处理，以及字号与坐标信息的提供上，而且在 PDF 文档的编辑、浏览、建立与打印上，easyPDF 与 Solid Converter PDF、iText 等软件相比功能并没有处于劣势。在图片萃取、表格处理、页面大小及排版上 easyPDF 都具有领先的优势。

图 3,图 4 分别为转换前的 PDF 文件和通过 easyPDF 系统转换后生成的 HTML 文件例图.

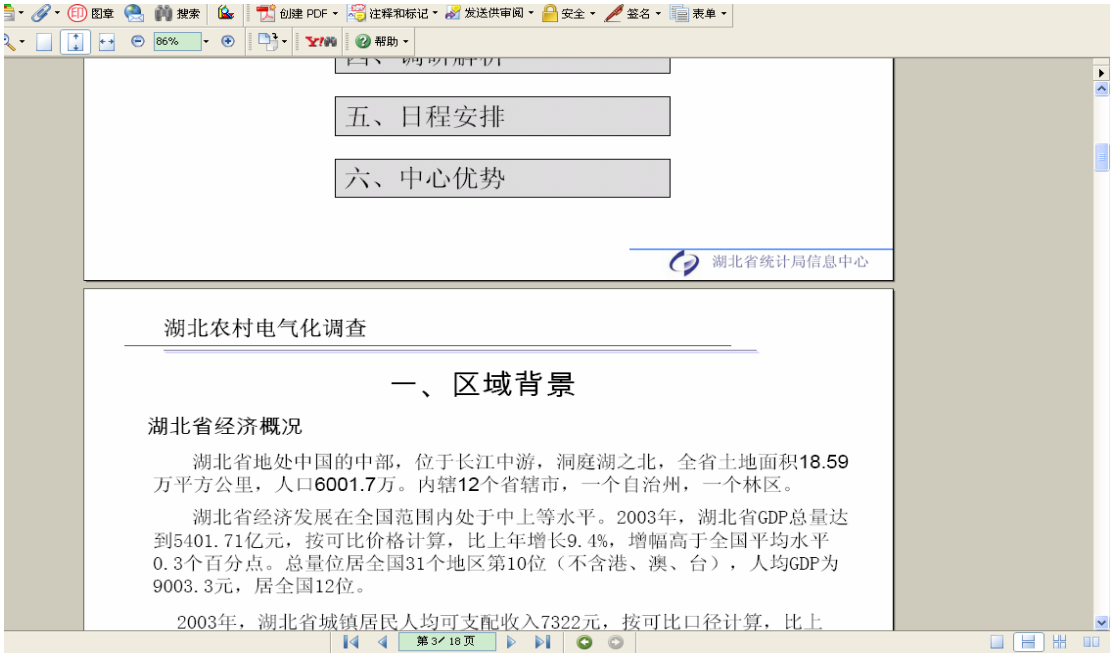

图 3　转换前的 PDF 文件图

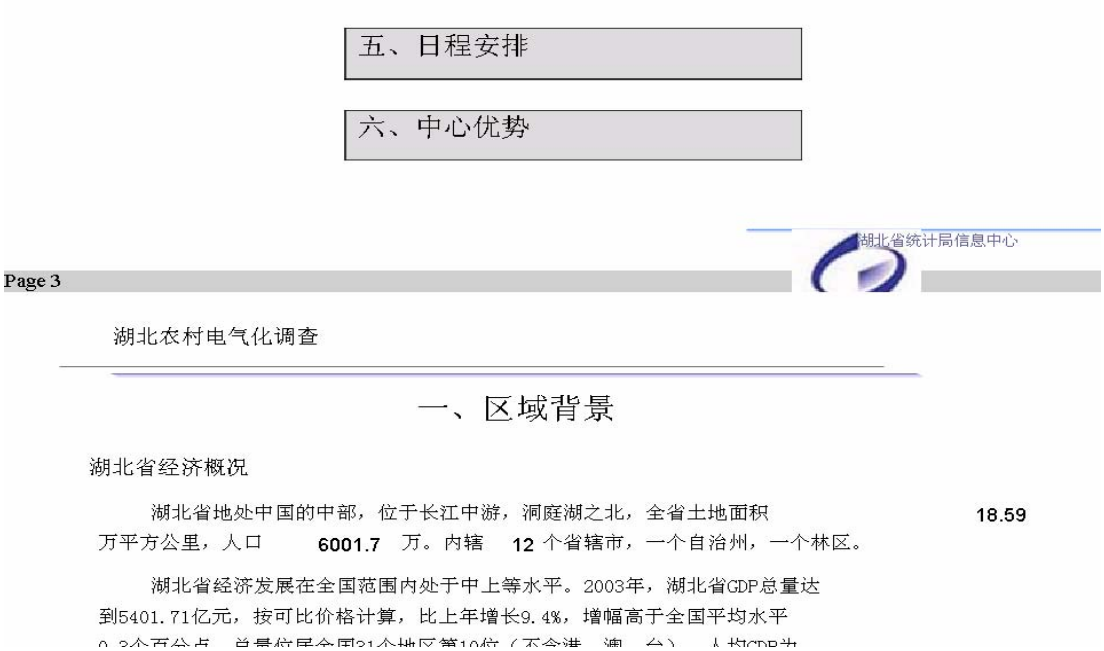

图 4　经过 easyPDF 系统转换生成的 HTML 格式例图

5. 总结

本文提出了一种纯 Java 的将 PDF 文件进行转换的系统.有良好的平台无关性和扩展行，并且由系统分析可见,虽然在 PDF 格式处理上,还存在这样那样的问题,通过本文提出的方案,我们可以提高 PDF 文档的增值应用,使 PDF 文件的内容更为方便的来为人所用，同时也可将转换运用到移动通信领域，带来经济效益。

参考文献

[1] Adobe Systems Incorporated. Adobe Portable Document Format Version 1.6, American Addison Wesley, 2004
[2] 蒋悦，吴壮志，赵旭林，怀进鹏. 基于文档树的 XML 文件转换.计算机工程，2003 年 12 月
[3] iText, a Free Java PDF Library - http://www.lowagie.com/iText/
[4] Multivalent - http://multivalent.sourceforge.net/
[5] PDFBox - http://www.pdfbox.org/
[6] PJX - http://pjx.sourceforge.net/
[7] XPDF - http://www.foolabs.com/xpdf/

A Novel Solution For Parsing And Transforming PDF File

Li Fuquan，Yang Jun，Man Chengyuan
School of Computer Science & Technology，Beijing University of Posts and Telecommunications，Beijing (100876)

Abstract:
This paper brings forward a solution to parse and transform PDF format files to XML files. Based on this scheme, PDF documents can be operated more conveniently. And from XML format, HTML and WML files can be transformed and sent by both web and wap. Compared with other solutions, easyPDF provides better results in pictures extraction and tables operation.
Keywords：PDF，ppXML format，easyPDF System，PDF Fonts，CMap

作者简介：
李福全，男，山东济宁人，北京邮电大学在读硕士研究生，计算机科学与技术专业，研究方向为计算机应用；
杨俊，男，四川成都人，北京邮电大学计算机网络中心副教授，硕士研究生导师。