办公文档与固定版式文档格式关系探讨

李 宁,田英爱,侯 霞,梁 琦
(北京信息科技大学计算机学院,北京 100101)

摘 要: 从文档承载信息的抽象程度,提出了文档分层的思想,分析了以流式办公文档和固定版式文档为主的不同层次文档之间的关系.利用 Tagged PDF,成功尝试了在固定版式文档中蕴含和提取办公文档信息,说明固定版式文档中容纳结构化办公文档格式的可行性,指出文档格式标准应贯通两种文档格式,形成完整的标准体系.

关键词: 文档格式; 文档处理; 标文通; 固定版式文档; Tagged PDF
中图分类号: TP317.1 **文献标识码**: A **文章编号**: 0372-2112 (2008) 12A-128-05

A Discussion on Relationship between Revisable and Non-revisable Document Formats

LI Ning, TIAN Ying-ai, HOU Xia, LIANG Qi
(Computer School, Beijing Information Science and Technology University, Beijing 100101, China)

Abstract: This paper brings forward a hierarchy of documents according to the abstract level of information being carried. Relations between different levels of documents were analyzed, centered on revisable office document and non-revisable document. An attempt to embed structural office document information into non-revisable document and to retrieve the information from it using Tagged PDF was experimented successfully. It shows the possibility and necessity to linkup the two kinds of document and form a consistent standard set.

Key words: document format; document processing; UOF; non-revisable document; tagged PDF

1 引言

电子文档作为当代的信息交换媒介,走过了长期的发展历程,正在发生着巨大的变化,例如:文档承载的内容不断丰富;文档从单一用途转为多种用途;文档的利用方式和显现手段日趋多样,这些变化导致文档格式日趋复杂.文档格式发展的总体趋势是,早先以忠实记录内容和高效信息编码为主要目标,近年重点逐渐转变到如何准确表示复杂的结构化信息,如何共享文档内容,以及如何适应多种应用上来.这种趋势,一方面使得计算模式从以数据为中心转变成为以文档为中心[1];另一方面,多种多样的应用需求给文档的创建、存储和利用带来前所未有的挑战.

今天,随着文档重要性的日趋提升,文档格式再次成为人们关注的焦点.在办公文档领域,2006 年 OASIS 的开放文档格式 ODF(Open Document Format)通过了 ISO 的表决,正式成为国际标准[2].微软也于同年开始放弃其封闭的二进制 DOC 文档格式,先后将它制定的 Office Open XML Format(简称为 OOXML)提交 ECMA 和 ISO 申请成为国际标准[3].2002 年开始,我国开始了国家标准"标文通"(英文简称 UOF,Uniform Office Format)的研制,并于 2007 年正式成为国家标准[4].在固定版式文档领域,2008 年 1 月,ISO 批准 Adobe PDF 1.7 版本成为国际标准,即 ISO 32000-1:2008[5].

目前,我们正在研讨如何制定我们自己的固定版式文档格式标准.我们认为,必须把握文档技术的未来发展趋势,突破历史藩篱,贯通办公文档和固定版式文档,通过技术创新构建合理的、基于 XML 的文档格式标准体系,形成完整的办公软件和电子出版产业链条,使产业最大程度地受益.因此怎样设计合理的文档信息记录结构,制定这样的文档格式标准是我们亟需解决的问题.本文试就此问题进行初步探讨.

2 文档信息的层次

"言为意之记",无论如何变化,文档作为信息载体的本质并没有改变.我们可以按文档承载信息的抽象程度把文档分为如下层次,见表 1.

作为记录语言的书面文字,一般采用纯文本方式来

收稿日期:2008-08-27;修回日期:2008-11-04
基金项目:北京市教委科技发展重点项目暨北京市自然科学基金(No. KZ200810772017)

表 1　文档信息抽象层次

层次	技术与工具	典型格式	抽象度
数据模型	XML、数据库	XML/ DB	高
书面文字/ 混合文档	文本处理、专用内嵌工具	TXT/ CDF/ SMIL	中
带式样的文本/ 办公文档	办公软件	UOF/ ODF/ OOXML	中
固定版式文档与图像	电子出版、阅读器	PDF/ CEB/ SEP	低

记录逻辑内容. 从书面文字到建立数据模型,是一个为了信息处理的目的对文档信息进行抽象、提炼的过程. 这一步主要是从书面文字中获得有价值的信息,进而转变为计算机可以理解的数据模型,进行更深层次的加工和理解. 当前,从书面文字到数据,广泛采用以 XML 为基础的标注来实现. 例如,将书面文字标注出分词和语法成分,可以通过规则进行自然语言理解;通过对信件的标注,获得收信人、发信人、发信日期、主题内容等信息,进而转换成关系模型存储到数据库之中,以便进一步利用.

书面文字以下,是信息不断具象化的过程. 书面文字加入式样之后,将有助于清晰而直观地显现内容,利于人类阅读. 这个层次的文档,还包含了反映语义的式样信息,例如,章节、表格和列表,它们可以看作是书面文字的一种特殊表达形式,而不仅仅是感观上的式样. 另外,这类文档还包含了复杂的编辑语义,这是供编辑工具理解而使用的一种特殊信息模型,例如,章节等内容的嵌套层次,索引和链接关系,内容的前后阅读顺序,正文与批注的划分等等. 这些信息为文档编辑提供了很大方便,例如,改变一个章节的字体只需要设置章节的式样,而不需要对章节的每一处文字进行设置. 这些信息也有助于在线阅读,例如,可以在某些时候把所有批注信息隐藏起来等等. 这一层次大体上是办公软件的覆盖范畴,因此我们也将这类文档称为办公文档.

表 1 的最下层是按固定版式或图像方式表示的文档. 这一层对文档的显现内容描述得最为具体,一般会根据显现介质(如屏幕或纸张),将要显现的文字和其他内容定位到以像素点为单位的坐标位置. 因此这类文档不管如何显现都可以忠实地保持版面原貌,适于阅读,但不适合做编辑和修改,因为删除或添加任何内容都会改变原来的面貌,导致需要重新计算页面布局. 另外,这一层文档一般很少带有更高层次的语义信息(包括编辑语义).

介于固定版式文档和逻辑内容之间,还存在一种混合文档,其中可能包含上述各类文档内容,通过接口彼此共存在一个容器中. 这样的混合文档一般不注重页面版式的忠实再现,但是希望各种内容尽可能在多种显现介质上恰如其分地呈现出来.

由于文档格式主要关注的是办公文档和固定版式文档,我们这里重点讨论这两类文档.

3　办公文档的特点

从表 1 我们看到,办公文档处于高层的抽象信息表示和底层的具象信息表示之间,记载的信息最为丰富. 办公文档很容易转换成纯文本文档,但反之不然;办公文档也容易自动地或半自动地转换为固定版式文档,所以有着承上启下的作用. 办公文档一般按如下的概念来描述:

●元数据,关于文档的信息,主要用于信息检索和管理.

●式样,若干格式描述的组合,用于简化格式描述,可以被多次复用.

●书签,标识文档的某个位置,以便快速定位.

●超级链接,文档内容间的语义关联.

●对象,文档中包含的各种多媒体内容和外部数据.

●节,最大的排版单元,不同页面式样的文档内容形成不同的分节. 节的设置通过节属性来描述,包括:修订、页边距、纸张、页眉页脚、脚注尾注、网格、文字排列方向和填充、分栏等.

●段落,节以下的排版单元. 段落的设置通过段落属性来描述,包括:缩进、行距、对齐方式、制表位、边框底纹和排版禁则等.

●句,段落之下的排版单元,是具有相同式样的连续的文字. 句的设置通过句属性来描述,包括字体和字体效果及边框等内容.

●其他,修订、批注、列表、表格、文本框、图形等等.

办公文档是非固定版式的文档,其最大特点是它使用流式灌排的排版方式. 即每页上有哪些内容是由上一页的内容决定的,第一页内容"灌"满了,便流入第二页,第二页"灌"满了,再流入第三页 ……,直到所有内容排出来为止. 这种方式有一个天然的局限,就是难以避免跑版. 即某种条件下看到的文档页数或对象的位置,再另一种条件下会发生改变(例如多出一页或少去一页等等). 很显然,流式灌排的排版方式受版面绘制算法影响很大,而系统所用的字体、标点压缩方式、度量单位选取乃至小数点舍入均可累积误差最终导致跑版. 目前只有借助固定版式文档才能解决跑版问题.

4　固定版式文档的特点

固定版式文档是在显现方面描述得最为精确的文档形式. 固定版式文档一般按如下的概念来描述:

●文件描述对象,用来描述这个文件的标题,作者,时间等.

●组对象,文档内容的起始节点.

●页面集合,页面对象的集合.

●页面对象,包含如何显示该页面的信息,例如使用的字体,页面的大小,内容(如文字、图片、批注、活动对象),等等.

●活动对象,如链接,文字,声音,电影等.

●图片对象及字体对象.

●流对象.包含大量的二进制内容,一般经过压缩.

●数字对象.

●引用机制:随机、快速找到各个对象的位置.

由此可见固定版式文档的主要组成部分是页面和对象,不具有办公文档的丰富语义.由于固定版式文档把内容按坐标位置固定下来,因此可以有效避免跑版问题.另外,由于难以更改,安全性相对较高.现阶段主要用于文档的浏览和长期保存.此外,由于可以加入数字版权保护机制,也广泛用于电子读物.但是,固定版式文档的优势在某些场合可能成为劣势,比如,不便编辑修改,检索效率不高,难以提取有用数据与其他应用结合.固定版式文档当前面临的另一项挑战是,移动计算的兴起导致显示设备种类繁多,它们的大小、分辨率、色彩、交互方式大相径庭,一种固定版式文档往往难以满足多种显示设备的需要,而保存多套版式文件又浪费资源,因此越来越多的版式文件便按照给定显示设备的规格从其他类型的文档动态转换过来.

5 关于构建结构化固定版式文档的尝试

我们认为,不应孤立地看待各种文档格式,应该站在全局的角度把各种文档看作一个整体.只有这样,我们才能把办公软件产业和电子出版等产业联合起来,形成以 XML 为基础的、完整的文档信息处理产业链条.这里最重要的是各层次的文档格式之间需要衔接.亟待解决的问题是:怎样才能让机器读懂展现给人类读者的文档,使之可以将有用的信息提取出来,融入信息处理的整体过程?

这个问题的实质,是让低层的文档承载高层的信息.在书面文字中加入 XML 标记从而标识出高层语义的做法早已被人们所接受.如何在办公文档中表达高层语义信息呢?“标文通”已经给出了一个解决方法[6].在“标文通”中,通过“用户数据集”将“标文通”的格式文本与用户自定义 XML 数据绑定,使文档内容与格式既可融合又可分离;既适合“所见即所得”方式的编辑应用,又适合与其它应用系统的集成.由此,可以进一步设想,固定版式文档能否以此建立与办公文档的对应关系,从而形成结构化固定版式文档呢?为此我们进行了一个初步的尝试,即利用 Tagged-PDF,通过标记将“标文通”办公文档信息嵌入 PDF 固定版式文档,然后再从中提取办公文档信息从而形成完整的“标文通”文档,从而成功回答了上述的问题.

Tagged-PDF[7]是 PDF1.4 带来的一个最显著的革新.其原理是,允许一般的 PDF 内容增加标记(Tag),这些标记可以是系统预设的,也可以是用户自定义的.它们可以使用户将自定义的 XML 标记与 PDF 的内容关联起来,从而达到在固定版式文档中加入结构化信息的目的.

图 1 左侧是利用办公软件“永中 Office 2007”生成的“标文通”文档.右侧是该文档的简化的 XML 表示.从中可以看到,这个备忘录主要由 8 个段落的文本内容组成.我们通过 PDF 的应用编程接口将该“标文通”文档转换为带标记的 PDF 格式,转换的同时,将“标文通”文档的 8 个段落的文本节点所对应的 XPath 记录在 PDF 的标记中.在 Adobe Acrobat Reader 7.0 中看到的 PDF 文档如图 2 所示.

图1 “永中Office 2007”生成的“标文通”文档及其XML简化形式

图 2 中右侧窗口内的内容是 PDF 中增加的标记.每个标记记录了 PDF 各部分文档内容对应的“标文通”文档节点.例如,“电力维修”这一段文字对应的是“标文通”文档中的节点“/uof:UOF/uof:文字处理/字:主体/字:段落[5]/字:句[2]/字:文本串/text()”.在“高亮标记内容”的方式下,当鼠标点击上述标记,图 2 左侧窗口

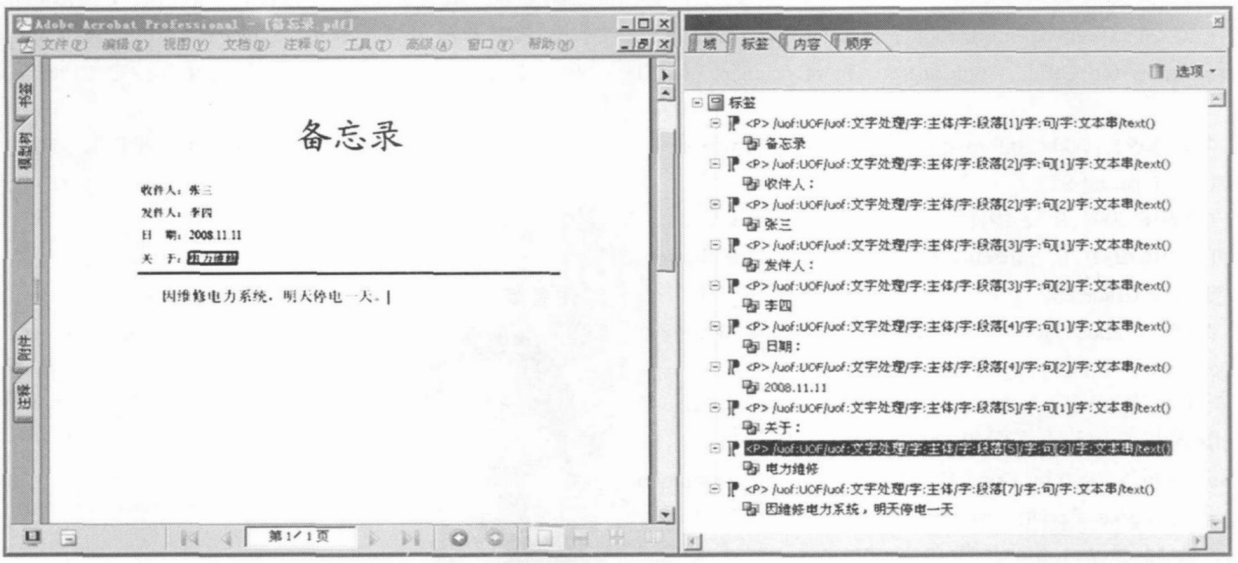

图2　转换为Tagged PDF的文档及其相应标记

中的相应文字被突出显示出来.

需要说明的是,在 PDF 中增加结构化信息已经有过先例. Hardy[8,9]等提出使用 XML 模板为已有的 PDF 文件增加 XML 结构化信息. 这种方式存在一定的困难,例如,需要按阅读顺序重新组织 PDF 内容,需要在字符一级进行内容的规则化,之后经过字符串一级的内容匹配才可能在 PDF 中建立结构化信息. 而办公软件大多具有固定版式文档的输出功能,在主动的固定版式文档构建过程中将结构化信息通过标签写入 PDF 并不存在上述困难.

为了逆向转换,我们需要保留"标文通"文档的框架,但是不需要保留具体的文本内容(用空文本节点替换). 当需要从 PDF 重构"标文通"文档时,我们仅需一段简单的程序,解析各个标记,用标记对应的文字内容按 XPath 替换"标文通"文档框架中的空文本节点即可. 这样便获得了"标文通"格式的办公文档,两种格式的文档内容由此形成有机关联.

在 Adobe 公司新的固定版式文档项目"MARS"中,对于结构化信息与版式信息的并存有了更好的方案[10]. MARS 文档格式沿袭了 PDF 的基本结构,但是采用 XML 语法表示文档格式,在图形等内容中支持 SVG 等 XML 描述规范,并使用 ZIP 压缩打包. 在 MARS 文档格式中,根节点"Document Catalogue"包含了一个称为"Structure Tree"的子树,专门用来存放 XML 结构化信息,在"Structure Tree"中各个节点可以通过标识符将 PDF 内容与 XML 元素进行对应. 这样我们便可把"标文通"表示的办公文档框架作为"Structure Tree"的内容存放在固定版式文档之中.

建立固定版式文档与办公文档的联系还有一种简单的方法,就是把各个层次的文档全部作为独立文件保留,打包成一个. 使用时,视应用的需要提取出最适合的文档来处理. 这种方法已经被一些开发单位所采用. 该方式固然简单,但是难以做到各个文档内容的同步一致,另外,各文档内容会有覆盖,造成资源浪费.

6　结论

综上所述,为了方便文档信息的交换和处理,设计出合理的文档信息记录结构,突破办公文档和固定版式文档的界限,从整体上构建文档格式标准体系是非常必要的. 虽然可以通过统一版面算法、增加分页参考标记等方法,减少办公文档的跑版问题,但是从根本上解决问题还需要借助固定版式文档格式. 文档格式标准的制定,应该打破现有格式的樊篱,使固定版式文档能够承载办公文档等更高层次的信息. 在固定版式文档中,我们利用 Tagged-PDF,将办公文档各内容节点所对应的 XPath 作为标记与 PDF 内容关联,实现了从固定版式文档中重构办公文档,进而获得更高层次的语义信息. 为制定结构化版式文档格式标准提供了参考.

由于 PDF 标准需要兼容以往存在的各类 PDF 文档,历史包袱很重,Tagged-PDF 的设计目的并非支持多层次的文档信息,其功能也有不少局限,例如对电子书的支持能力不足,所以 PDF 格式并不完美. 我国在办公软件和电子出版领域有深厚的技术积淀,曾开发出多类具有自主知识产权的文档格式,并拥有庞大的文档处理应用市场,建立完善的文档标准体系,构建完整的文档处理产业链条是非常可行的. 希望本文的讨论能够有助于这个目标的实现.

参考文献:

[1] Weck W. Document-centered computing: compound document editors as user interfaces[J]. Journal of Symbolic Computing,

1997,11:1 - 24.

[2] ISO/IEC 26300:2006, Information technology - Open Document Format for Office Applications (OpenDocument) v1.0 [S].

[3] ISO/IEC 29500:2008, Information technology - Office Open XML file formats[S].

[4] GB/T 20916-2007, 中文办公软件文档格式规范[S]. GB/T 20916-2007, Specification for the Chinese office file format[S]. (in Chinese)

[5] ISO 32000-1:2008, Document management-Portable document format - Part 1:PDF 1.7[S].

[6] 李宁,牟永敏,董慧,方春燕. 文档格式中"内容"与"表现"的分离与融合[J]. 电子学报,2007,35(2):375 - 378.
Li Ning, Mu Yong-min, Dong Hui, Fang Chun-yan. Separation and combination of content and appearance in document format [J]. Acta Electronica Sinica, 2007, 35(2):375 - 378. (in Chinese)

[7] Adobe Systems Incorporated. PDF Reference (Second Edition) Version 1.3[M]. Boston:Addison-Wesley, 2000.

[8] Hardy M R B, Brailsford D F. Mapping and displaying structural transformations between XML and PDF[A]. ACM Symposium on Document Engineering[C]. Virginia:ACM Press, 2002. 95 - 102.

[9] Hardy M R B, Brailsford D F, Thomas P L. Creating structured PDF files using XML templates[A]. ACM Symposium on Document Engineering[C]. Wisconsin:ACM Press, 2004. 99 - 108.

[10] Hardy M R B. The Mars project - PDF in XML[A]. ACM Symposium on Document Engineering[C]. Manitoba, ACM Press, 2007. 161 - 170.

作者简介:

李 宁 男,博士,研究员. 1964 年生于北京. 北京信息科技大学计算机学院副院长,全国电子政务标准化总体组中文办公软件基础标准工作组副组长. 研究方向:文档处理、XML 技术与应用.

E-mail:ningli@public2.bta.net.cn