开放的复合文档OpenCDF模型及其实现

侯霞　李宁　樊凯

（北京信息科技大学计算机学院侯霞北京 100101）

**摘要**　随着资源多样化和信息集成化程度的提高，文档需要集成不同来源、不同类型的信息。开放的复合文档OpenCDF被定义为一个文档容器，其中包含若干个微文档对象和一个文档框架。组成文档的微文档对象可符合不同的文档标准和规范，通过文档框架建立逻辑关系，在容器中集成为一个逻辑文档，并借助文档框架进行互操作。提出OpenCDF模型及其基于MVC的处理架构，并通过Silverlight等技术实现一个基于WEB的文档处理系统说明了的本模型和架构的可行性。

**关键词　MVC**，微文档对象，开放的复合文档，文档处理

**中图法分类号**TP317.1　　　**文献标识码**A

The Model and Implementation of Open CDF

HOU Xia　LI Ning　FAN Kai

(Computer School, Beijing Information Science and Technology University, Beijing100101,China)

**Abstract**: A document need to integrate information from different resources and formats, because of the more diversification of resources and higher degree of information integration. Open Compound Document Format (OpenCDF) is introduced. An OpenCDF document is a container, composed of zero or more microdocument objectsand a document frame. Each microdocument object complies with certain document standard or specification, The frame describes the logical relations among these microdocument objects and information about interoperability. The model of OpenCDF is presented and a prototype system based on MVC is implemented to illustrate feasibility of OpenCDF.

**Keywords**MVC, Microdocument Object, Open Compound Document, Document Processing

# **概述**

随着国际化趋势的增强，文档的互操作需求日益显著。基于XML的文档格式替代二进制文档成为不可逆转的趋势。目前，出于不同的目的和用途，不同的组织研究发布了不同的文档标准。例如微软的OOXML [1]、OASIS的ODF[2]和中国的UOF[3]等都是针对办公软件的文档标准。基于XML的文档标准的发展对文档的互通意义重大，但是不同标准之间存在着大量差异，使得不同文档标准之间的互操作仍然存在问题。

随着资源多样化和信息集成化程度的提高，用户需要将各种不同来源、不同类型的信息集成在一起形成一个复杂的文档，称之为复合文档，并完成文档内部和文档之间的互操作。为了使文档不再仅限于文字内容，而成为一个多种类型内容的容器，多种技术被开发并成功应用于处理复合文档。例如OLE (Object Linking and Embedding)，COM(Component Object Model)，SOM(System Object Model)技术，OpenDoc 等等。这些技术为复合文档的处理提供了良好的平台支持。若干复合文档架构在这些技术的基础上得以实现[4-6]。现有的各种主流Office套件也是利用类似技术来支持文档中不同类型的内容。

上述技术都是基于面向对象技术，对不同类型的对象进行处理。以此，可将异构的对象集成在复合文档中，可扩展以字处理为主的文档编辑环境的包容性，提高对象处理的灵活性。此类技术为复合文档提供了实现基础，能够在一定程度上提供文档的互操作。但是其实质是采用不同的软件处理不同的内容，没有统一的描述格式，文档之间缺乏有机的联系。这些仍是互操作的障碍。

W3C的复合文档（Compound Document Format，CDF）[7]为文档互操作提供了一个新的解决途径。CDF类似于一个文档容器，将多种基于XML格式的文档（例如XHTML、SVG、SMIL和XForms等）结合在一起，，同时通过文档之间的API有效地将文档联系在一起。CDF克服了使用面向对象技术支持复合文档的缺陷。然而，CDF中如何使开发工具将不同命名空间的文档融合在一起，并能够适应schema的变化以及适应多种浏览器也是一项挑战。

在文献8中，作者提出一种基于MVC的文档处理框架。在该框架中，文档仅做为一个包含若干“微文档”的容器；每个微文档可针对某类文档对象（文档片段），不同的“微文档”可以独立存在或通过文档容器形成复合文档。复合文档的处理将分解为各类微文档的处理，而不同微文档可通过不同的途径和手段来处理。该框架充分利用了MVC架构的优势，可有效提高文档处理软件的可重用性和灵活性。

为了与W3C的复合文档相区分，本文将文献8提出的复合文档称为开放的复合文档（Open CDF），并在文献8的基础上，设计了一套Open CDF的模型，详细定义了其处理框架的设计思路和实现方案。

# **相关技术**

本文主要利用下述技术实现了一个基于WEB的文档处理原型系统。

## **Silverlight**

微软Silverlight是一个跨浏览器、跨客户平台的技术，能够设计、开发和发布有多媒体体验与富交互(RIA,Rich Interface Application)的网络交互程序[9]。

Silverlight将多种技术结合到一个开发平台，可以在其中选择符合需求的合适的工具和编程语言。它能够与可以与已经存在的JavaScript和ASP. NET AJAX代码无缝集成，不会使已经创建的功能缺失。

SilverLight的一大特性是包含了Windows Presentation Foundation(WPF)技术，这个技术在创建用户界面时极大的扩展了浏览器元素。WPF可以创建融合图形、动画、媒体和其他的富客户端特性，扩展了基于浏览器的用户界面，超越了HTML所提供的。

## **XAML**

可扩展应用程序标记语言(eXtensible Application Markup Language，XAML)提供了创建WPF元素的声明性标记[10]。XAML是微软公司为构建应用程序用户界面而创建的一种新的描述性语言。XAML提供了一种便于扩展和定位的语法来定义和程序逻辑分离的用户界面，而这种实现方式和ASP.NET中的"代码后置"模型非常类似。由于XAML是一种基于XML的解析性语言，所以它本身就是一个组织良好的XML文档，相对于HTML，它的语法更严谨、更明确。

## **XSLT**

XSLT是扩展样式表转换语言（Extensible Stylesheet Language Transformations）的简称[11]。XSLT是把XML文档转化为另一文档的转换语言，即将源文档的所有数据或者部分数据，利用XPath进行选择，生成另外的XML文档或者其他可直接显示或打印的文件格式（例如 HTML文件、RTF文件或者TeX文件等）。XSLT本身也是一份XML文档，所以它也必须遵守严格的XML规范。

XSLT语言是声明性的语言，即XSLT程序本身只是包含了一些转换规则的文档。而这些规则可以被递归地应用到转换过程中。XSLT处理程序会首先确定使用XSLT中的哪些规则，然后根据优先级作出相应的转换操作。

# **OpenCDF模型**

## **基本模型**

本文提出开放的复合文档Open CDF模型，本文给出如下定义。

**定义1“复合文档”**：一个文档容器，是由若干个微文档对象和一个文档框架组成的逻辑文档。由于该容器对其可容纳的微文档对象类型和标准不做限制，本文称其为开放的复合文档，后面简称OpenCDF。在本文后面的部分，如果没有明确说明，则“文档”即OpenCDF。

**定义2“微文档对象”**：符合某种文档规范或标准的文档片段或整体，是OpenCDF的组成部分，其来源、类型不限。各微文档对象可独立存在。对OpenCDF的处理实际是分解为对各微文档对象的处理。

**定义3“文档框架”**：描述微文档对象在文档中的组织结构，类似于文件系统的索引。其中规定了各微文档对象在文档中的逻辑关系和组织顺序、各“微文档对象”的类型、微文档对象及其对应物理文档之间的关联等，还描述一些关于文档的元数据、微文档对象之间的互操作信息等数据。具体实现时可以是一个物理XML文件。

图1是一个OpenCDF模型的示例图。该文档包含一个符合UOF标准的字段落微文档对象、一个符合OOXML标准的表格微文档对象、一个符合ODF标准的列表微文档对象，等等。



图1 OpenCDF示例图

## **文档框架模型**

图2用Schema的形式给出了文档框架的模型。其中

* 元数据：用于描述文档的作者、创建时间、该文档中包含的微文档对象个数等信息。其内容可根据需要扩展。
* 主体：记录了OpenCDF逻辑文档的组成模型。目前主要借用了UOF中节和段落的概念。段落和节都是逻辑概念。每个段落针对一个实际的微对象，若干段落组成一个节。段落和节都可设置各自的格式属性。
* 微文档对象集：一个OpenCDF文档所包含的微文档对象都记录在微文档对象集中。
* 操作集：一个OpenCDF文档所包含的操作都记录在操作集中。

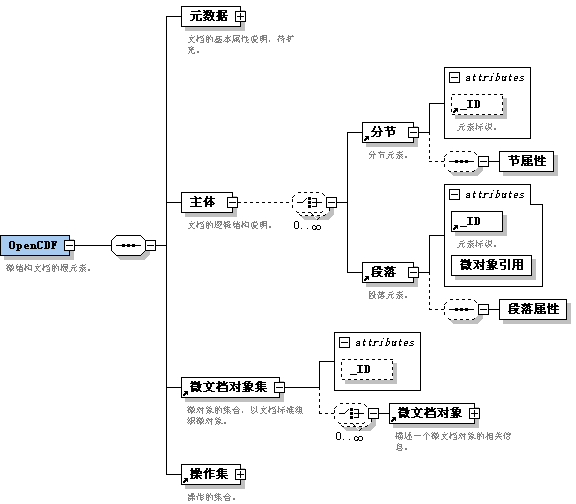


图2文档框架模型

## **微文档对象模型**

微文档对象是文档中的处理粒度，通过几个属性来描述：

* ID：每个微文档对象具有唯一的ID标识，可在文档主体的段落中被引用；
* 相对路径：每个微文档对象虽然是文档的组成部分，但是它的来源可以是一个独立存在的物理文件。该文件与文档框架所在物理文件的相对路径，记录在其微文档对象的“相对路径”属性中，供OpenCDF集成时使用；
* 类型：微文档对象的“类型”标识该对象所符合的文档规范或标准。对供OpenCDF的处理实际是对某个微文档对象的处理，处理时先要识别该对象的类型，然后调用不同的处理控件或处理方法。例如一个微对象的类型是“OOXML”，则处理该对象时可调用OOXML的控件。

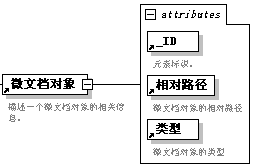


图3 微文档对象模型

## **操作模型**

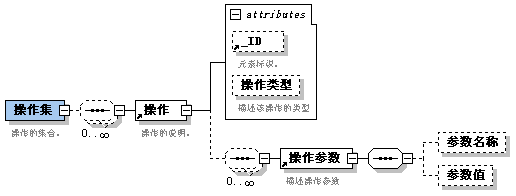


图4 操作集模型

操作集包含0至多个操作，用于记录互操作的信息，即可用于从微文档对象中提取，也可用于向微文档对象中输入。对于操作的描述如图4所示，其中每个操作包含：

* “ID”属性：用于唯一标识该操作。
* “操作类型”属性：用于指定操作的种类，如“格式刷”就是一个操作的例子，需要从一个微文档对象中提取格式信息，然后复制到另一个微文档对象。
* 0至多个“操作参数”：用一系列参数名称－参数值对记录互操作中涉及的参数和参数值。个数根据需要扩展。

# **微对象文档处理架构的设计**

## **文档架构**

本文采用了文献8中所提的文档处理架构，采用分而治之的方式，将对于文档的处理分解为对各个微文档对象的处理，以降低处理软件的复杂度。文献8中分析了文档处理与MVC架构中各部分的对应关系。在对各个微文档对象进行处理时，采取MVC架构，可充分利用MVC的优势，有效地提高软件的可重用性和灵活性。

本文在文献8的基础上，进一步分析了文档处理中MVC架构的实施方案，将对微文档对象的处理分别对应到模型、视图和控制器三部分。本文主要利用XSLT、Silverlight和XAML技术实现了一个基于WEB的文档处理原型系统，用以验证本文所提模型和框架设计的合理性与可行性。图5给出了OpenCDF处理架构图。



图5OpenCDF处理架构

* 模型：用于描述每个微文档对象的结构，符合该微文档对象所遵从的文档规范或标准。例如一个UOF段落微文档对象的模型符合UOF中对于段落的定义模型。
* 视图：采用HTML和Sliverlight技术实现用户界面，用于文档的浏览和与用户的交互。可用于呈现若干微文档对象集成后的整个逻辑文档的浏览结果。
* 控制器：适用于对微文档对象进行处理。根据功能不同又可分为浏览控件、编辑控件和互操作控件。

## **文档的处理**

文档的处理被分解为对各微对象的处理，处理功能分布在不同类型的控件中。本部分给出各种处理的流程和实现方案设计。

### 文档浏览

虽然对文档的处理被解析为各个微文档对象的处理，但是做为一个容器，文档将各微文档对象集成在一起时需要为用户提供一个统一的浏览界面。由于各个微文档对象可以是异构的，即来源和格式不同，因此整个文档的浏览需要对各个微文档对象进行处理、转换，最终提供统一的用户浏览界面，这些工作是由浏览控件完成的。图6给出了一个文档在WEB浏览器中的浏览结果，是图1示例的一个具体实现结果。

文档浏览的处理过程和实现方案如下：

1. 解析主控文档，获取文档逻辑结构信息及微文档对象的具体信息。
2. 依次对各个微文档对象进行解析，并通过XSLT转换为XAML文件，一一对应。
3. 根据文档框架中的逻辑结构信息，将2)中得到的各个XAML文件按照文档结构进行组装，生成对应整个文档的XAML文件。
4. 通过Silverlight技术动态调用最终的XAML文件，将文档内容信息和格式信息在浏览器中呈现给用户。

符合ODF标准的列表微文档对象

符合UOF标准的字段落微文档对象

整个复合文档的页边距。

符合OOXML标准的表格微文档对象

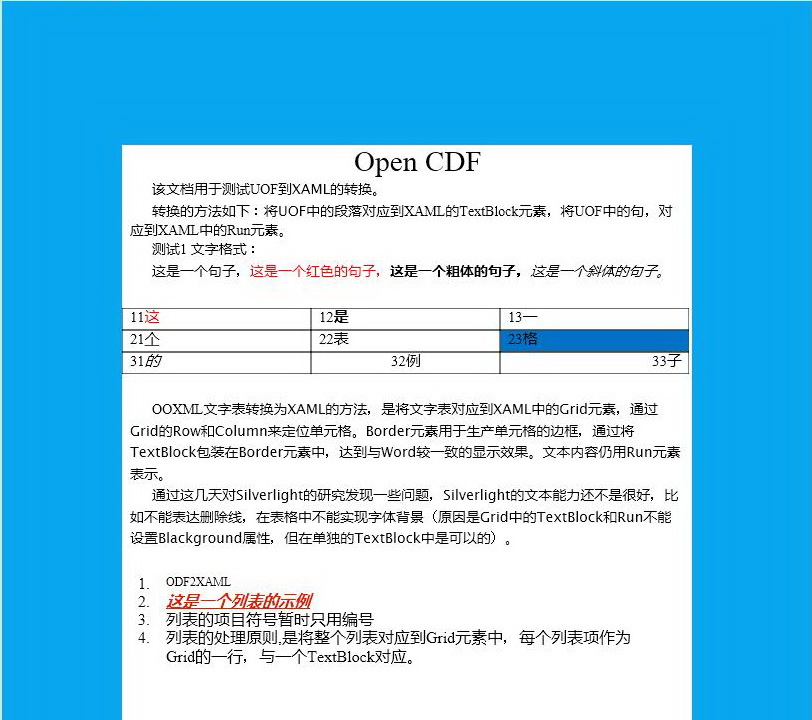


图6 OpenCDF文档的浏览界面示例

### 文档编辑

文档编辑是针对各个微文档对象分别进行的。例如，文档浏览过程中，可选定一个段落，即一个微文档对象，进行编辑操作。编辑可调用现成的控件，也行自行处理。下面以段落为例说明文档编辑的处理流程和方案：

1. 用户点选段落，并请求编辑。JS响应用户在Silverlight中的点击操作，获取该段落的ID，并将该ID作为Request的参数发送给服务器。
2. 服务器接收编辑请求，根据Request传递的ID参数，在文档框架中定位目标编辑段落所对应的微文档对象信息。
3. 根据微文档对象的类型，调用不同的文档编辑控件，并为用户提供编辑界面。例如对于一个符合OOXML标准的表格微文档对象，可以调用相应的Office控件在WEB浏览器中显示Office编辑界面。
4. 当用户完成编辑或取消编辑时，服务器根据用户提交数据更新微文档对象的模型，并调用文档浏览组件，呈现更新后的结果和内容。具体处理流程可参看文献8。

### 文档互操作

各个微文档对象在容器中集成为一个逻辑文档，这种形式易于使本来各自独立的微对象产生一定的关联，进而需要一定程度的信息互通。同时文档从静态向动态、智能发展的趋势，也需要“微对象”之间提供互动的能力。本文将这种整体与部分，部分与部分之间信息的互通互动能力称为文档互操作，目前这种互操作分为两种：

* 一种是文档框架和微文档对象之间的操作。例如整体文档的浏览。
* 一种是微文档对象之间的操作。例如格式刷，将一个微文档对象的格式复制到另一个上面。

如果允许微文档对象之间的直接交互，则需要针对每一类微文档对象的每一类操作指定交互准则，如接口等。这样显然缺乏灵活性和可扩展性。此外，如果允许微文档对象之间直接交互，则文档框架的作用将被限制在仅提供文档逻辑结构的范围内。因此，本文将微文档对象之间的操作进行抽象，由文档框架来记录抽象化的操作，并置于操作集中。由此，微文档对象之间的互操作演化成“微文档对象—文档框架—微文档对象”的操作。这样，每个微文档对象只要关心与文档框架的交互即可，而无需关心其他微文档对象的任何信息。不但有助于降低互操作的难度，而且简化了文档交互接口的复杂度并具有良好的可重用性。

表1 sdsadadadsa

|  |  |  |
| --- | --- | --- |
| 1 | 2 | 3 |
| 4 | 5 | 6 |

表1用于测试的数据。

互操作处理流程较为复杂，涉及文档框架和微文档对象的交互。本文以格式刷为例说明微文档对象之间的互操作处理。

1) 用户点击进入格式刷操作界面。

2) 用户选择段落，并开始“复制格式”或“使用格式”。

3) 根据用户所选段落的ID，由文档框架定位段落。

4) 根据微文档对象的不同类型，调用该类型文档的格式处理控件（该控件实现了格式刷操作的API），对该段落的格式信息进行相应操作，包括提取微文档对象的格式信息存储到文档框架。

5) 处理程序从文档框架提取格式信息存储并应用到目标到微文档对象。

5) 操作完成后，保存文档（包括文档框架和相关微文档对象），并返回格式刷操作界面。

6) 退出格式刷操作，返回文档浏览界面。

此时格式的更改也属于对微文档对象的编辑，编辑结果更新后相应的浏览内容也要同步更新。

对于本格式刷的例子，可将相应操作设置如下：

<操作ID=”O\_001”type="格式刷">

<操作参数>

<参数名称>中文字体</参数名称>

<参数值>宋体</参数值>

</操作参数>

<操作参数>

<参数名称>字号</参数名称>

<参数值>10</参数值>

</操作参数>

……

</操作>

# 结束语

随着资源多样化和信息集成化程度的提高，文档需要集成不同来源、不同类型的信息。W3C的CDF为复合文档的集成提供了一个很好的思路和模式。但是CDF中需要将不同命名空间的文档融合在一起，并能够适应schema的变化以及适应多种浏览器也是一项挑战。这在CDF的实际应用中一个很难解决的问题。

本文在文献8和CDF的基础上，提出一个新的文档模型－开放复合文档Open CDF，其中复合文档只是一个文档容器，包含若干个微文档对象和一个文档框架。组成文档的微文档对象可符合不同的文档标准和规范，通过文档框架建立逻辑关系，在容器中集成为一个逻辑文档，并借助文档框架进行互操作。本文提出了一个较为详尽的Open CDF模型，及其基于MVC的处理架构，通过利用XLST、Silverlight和XAML技术实现了一个基于WEB的文档处理系统，说明了本文研究内容的合理性和可行性。OpenCDF可以将多标准的文档融合在一起，通过分而治之的理念降低文档处理软件的复杂度，并为用户提供统一的文档浏览界面。

本文工作提出了Open CDF的理念，仅是对其进行了一个有效的初步尝试。在模型设计一些方面还有待完善。例如对互操作的模型设计、主体中逻辑节和段落的属性使用等将是今后需要深入研究的一些内容。

# 参考文献

1. ISO/IEC 26300:2006, Information technology -- Open Document Format for Office Applications (OpenDocument) v1.0[S].
2. ISO/IEC 29500:2008, Information technology -- Office Open XML file formats[S].
3. GB/T 20916-2007, 中文办公软件文档格式规范[S].
4. G.H. ter Hofte, H.J. van der Lugt. CoCoDoc: a framework for collaborative compound document editing based on OpenDoc and CORBA [A]. Proc. of the IFIP/IEEE international conference on Open distributed processing and distributed platforms，Toronto, Ontario, Canada. 1997:15-33.
5. F. Moelaert El-Hadidy, W.B. Teeuw and H. Bakker. An Innovative Approach for Designing Collaborative Applications using OpenDoc: from theory to practice [A]. Proc. of 8th Conference on Software Engineering Environments. Cottbus, Germany, Apr 1997:42-52.
6. I.Satoh. A Compound Document Framework for Multimedia Networking[A]. Proc. of 1st International Conference on Distributed Frameworks for Multimedia Applications. 2005:80-87.
7. T. Mehrvarz, L. Pajunen, and J. Quint et al. Compound Document by Reference Framework 1.0. [2010-8-20]. http://www.w3.org/TR/CDR/.[EB/OL].http://www.w3.org/TR/2007/CR-CDR-20070718/
8. X. Hou, N.Li, Y. A. Tian. A Framework based on MVC of Document Processing [A]. Proc. of 2009 COINFO. Beijing, China, 2009:290-294.
9. Wikipedia. Microsoft Sliverlight. <http://en.wikipedia.org/wiki/Microsoft_Silverlight>
10. XAML Overview (WPF). <http://msdn.microsoft.com/en-us/library/ms752059.aspx>
11. 维基百科. XSLT.http://zh.wikipedia.org/zh/XSLT