基于 XQuery 的 ODF 文档的数据集成

Data Integration of Open Document Format on XQuery

孙清国[1,2]， 朱　玮[1]， 刘华军[3]， 张　鹏[3]

（1 中国科学院长春光学精密机械与物理研究所 吉林长春 130033；
2 中国科学院研究生院 北京 100039；3 91550 部队装备部 辽宁大连 116023）

摘　要： 随着 ODF 标准的日益推广，越来越多的文档尤其是办公文档采用 ODF 格式，对 ODF 文件的数据集成变得越来越重要。众多的应用系统支持统一的文档标准，也带来了数据集成的便捷性。本文提出了一个基于 XQuery 语言，对 ODF 文档的数据集成框架，研究了对 ODF 文档进行数据集成的一些关键问题。这种方案具有一定的扩展性，只要开发相应的适配器就可应用于其它类似的标准，如 OOXML 和 UOF 文件标准。

关键词： ODF XQuery XQJ 数据集成 Saxon

1 引言

数据集成是企业信息系统的核心部分之一，这对于实现 EAI（企业应用集成），进行企业内部整合具有极其重要的意义。而对于企业内部大量的办公文档的数据集成同样变得越来越重要。目前，市场上存在多种办公软件，如微软 Office、Sun 的 Star Suite、红旗中文 2000 的 RedOffice、金山 WPS、永中 Office 等等。不同厂家采用了不同的文档格式，对这些不同格式的文档进行数据查询和数据集成是相当困难的。

在 ODF 开放文档标准出现之后，我们可以看到，不管在一些 Office 工具也好，还是网络上的一些应用也好，还有很多在电子政务、电子商务的应用中都可以共享到同样的一些文档。在越来越多的应用系统支持 ODF 开放文档标准的大潮下，对这些文档进行数据集成有了一定的简易性。只需要对 ODF 标准文档格式进行解析、数据集成，即可实现不同应用的办公文档的数据集成。

ODF，全称为 Open Document Format，开放文档格式是一个基于 XML 的开源文件格式，用来存储和转换文本、电子数据表格、图表以及陈述文件。文件以 ODF 格式保存，称作“开放文档”。因为 ODF 是基于 XML 的文件格式，那么我们可以利用 XQuery 查询 ODF 中的数据，并进行一定的数据集成。本文根据 ODF 文档的格式，采用 XQuery 查询语言，提出一个基于 XQuery 的 ODF 文档的数据集成框架，解析其系统结构，并研究了相关的实现技术。这种方案具有良好的实用性与扩展性。

2 关键技术

2.1 ODF 格式解析

ODF 格式是基于 XML 的纯文本格式，这与传统的二进制格式不同。ODF 格式最大的优势在于其开放性和可继承性，基于 ODF 格式的文档在许多年以后仍然可以为最新版的任意一款办公软件打开使用，而传统的基于二进制的封闭格式的文档在多年以后可能面临的问题是：由于办公软件的升级或者原先的办公软件公司的倒闭导致老的文档不能够再使用，显然这对于用户而言将面临可怕的数据损失，这是用户所不能容忍的，而开放的 ODF 格式将很好的解决这一问题。

ODF 作为标准文档格式，由 OASIS 负责制定，它的目的是改变目前办公软件相互封闭、文档格式互不兼容的糟糕情况。ODF 格式可以让不同程序、平台之间都自由的交换文件而不需要理会是何种应用程序所产生的文件，其主要的支持厂商是 Corel、IBM、Opera、甲骨文、红帽以及国内软件厂商中文 2000 公司等。微软公司同时已经推出的 Word、PowerPoint 和 Excel 之间转换器，支持开放文件格式（ODF）。微软的文件格式转换器，可以在 SourceForge. net 网站下载，它允许用户使用 XML 格式或开放文档（ODF）打开和存储文件。

ODF 格式在 2006 年 5 月已经通过 ISO 批准，正式成为国际标准，标准号为：ISO/IEC 26300。ODF 文档是基于 XML 语言的纯文本文，具有容易辨认的扩展名，和微软的文档格式. doc 或. xls 相类似。遇到的最普遍的扩展名包括：

. odt 用于字处理文件

. ods 用于电子数据表格

. odp 用于陈述文件

. odg 用于图形文件

. odf 用于公式或其他的数学方程式

一个 ODF 文档实质上是一个打包的文件，并且通常都经过了 zip 格式的压缩。我们完全可以用现有的任意一款压缩软件将 ODF 文件解压，查看其里面的内容就会发现其本质。一个 ODF 文件解压后会得到一个与原文件名相同的文件夹，该文件夹里面一般会含有以下子文件夹与文件：

Configurations2 – – 文件夹

META – INF – – 文件夹

Pictures – – 文件夹

Thumbnails – – 文件夹

content. xml – – 文件

meta. xml – – 文件

mimetype – – 文件

setttings. xml – – 文件

styles. xml – – 文件

表 1 ODF 文本文档文件

文 件	描 述
content. xml	包含所有的文档文本以及索引标记、样式信息的链接等等。该文件是文档的主体。
meta. xml	包含文件元数据，比如说作者和文档标题。
styles. xml	定义文本的格式，比如字体的更改、段落方向、页面样式等等。ODF 会尽可能地保持样式与内容分离，因此 content. xml 文件中不会混入任何此类信息的说明。最多是含有一些从内容到样式的链接。
Thumbnails/ thumbnail. png	提供文档第一个页面的缩略图。
mimetype	这个是唯一的非 xml 文件，它只是说明了该 odf 文件的类型。
META – INF \ manifest. xml	这个文件在 META – INF 目录中，它列出了 odf 文件的压缩包中所有的文件和目录集合。
settings. xml	该文件列出 odf 文件的所有设置，就像一个配置文件。例如，列出了文档页面的长宽等信息。

正如一些设计相当良好的 XML 格式，文档文件并不是特别难于理解。获得更多信息那么可以阅读参考资料[1]。表 2 提供了一些示例。

表 2 XML 标记

Tag	Description
< office: document – content >	根标记。注意，所有的 XML 名称空间（包括 office）都定义在这个标记中。
< office: font – face – decls >	包含文档中所使用的字体。
< office: automatic – styles >	包含最基本的样式。styles. xml 文件对这些样式作出了详细说明。
< office: body >	包含文档的主体。
< text: p >	对应于 HTML 中的 < p > 标记，出现在整个段落的两侧。
< text: span >	对应于 HTML 中的 < span > 标记，允许我们为段落了某些特定的部分指定样式。

文档的实际内容在 content.xml 中,那么我们可以通过 XQuery 查询其中我们想要的数据。微软的 OOXML(Office Open XML File Formats)标准和我们中国提出的 UOF(Unified Office document Format)标准拥有类似的特点,本文不作更多叙述。

2.2 XQuery 与 XQJ

随着存储在 XML 文档中的信息量的增长,能有效并且高效的存取 XML 的信息相应的也变得越来越重要。要做到这点,必须要有一个能够准确获得所需信息、更新 XML 数据源中数据的可表达的查询语言。XQuery 正是这样的语言。

XQuery[2] 是一种 XML 查询语言,由 W3C 定义及标准化。XQuery 是以 XQuery 1.0 以及 XPath 2.0 数据模型的形式定义的,并将 XML 文档的分析结构描述为有序的,做上标记的树,树上的每一个结点都有一个不同的身份并可能具有简单的或者复杂的类型。XQuery 能够被用于对没有任何模式(schema)的 XML 数据进行查询,也可以对由 World Wide Web Consortium (W3C) XML 模式或者由文件类型定义(DTD)来管理的数据进行查询。

XQuery 作为一种功能强大的 XML 数据查询语言,XQuery 可以实现以一种基于标准接口的统一模式查询。由于 XML、XQuery 所具备的种种优势,目前已经有越来越多的研究项目采用 XML 作为信息交换模型,XQuery 作为数据查询工具,以 XML 统一视图实现各种异构数据源的信息集成,突现综合信息查询、处理。

XQuery 查询是用于查询 XML 的声明性语言,类似于用于查询关系数据的 SQL。大部分 Java 程序员熟悉 JDBC,它提供了标准的 Java API 来和各种关系数据库的 SQL 引擎进行交互。XQJ 具有相同的目的:它向 Java 程序员提供了标准的 Java API 用于和各种 XML 数据源的 XQuery 引擎进行交互。

XQJ 也被称为 JSR 255[3],因为它是由 Java Community Process 设计的。这个 JSR 255 规范定义了接口和 Class 集合,用于 Java 应用能够向一个 XQuery 引擎对一个或者多个 XML 数据源提交 XQuery 查询和使用查询结果。

因为 XQJ 目前只是一个草案,所以我们的系统采用 XQJ 的一个 JAVA 开源实现 SAXON 来实现。

3 系统结构

本文提出的数据集成框架采用的是面向对象的 Java 和 XQuery 技术,其集成框架结构如图 1 所示。整个集成框架采用 3 层体系结构,从上到下分别是:客户端层、查询引擎层和数据源层。

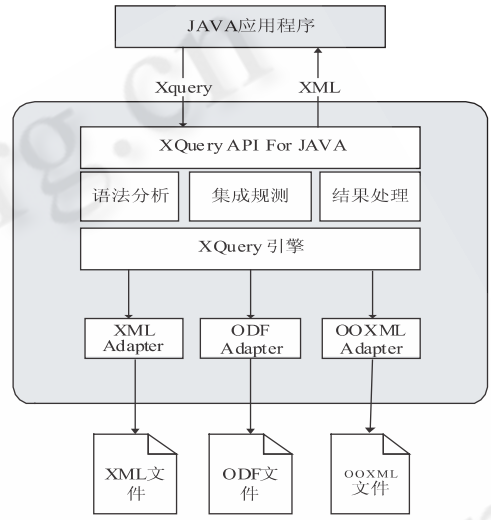

图 1　基于 XQuery 的文档数据集成框图

客户端层:由于采用 JAVA 语言实现,则客户端可以是 JAVA 应用程序、JSP、SERVLET 和 APPLET 等等。JAVA 客户端主要用来发送 XQuery 查询请求,并接收查询引擎返回的 XML 结果,并根据具体情况将 XML 结果转换成各种格式,比如 HTML,用于输出。

数据引擎层是该集成框架的核心层,它包括 XQJ 接口、语法分析、集成规则、结果处理、XQuery 引擎和适配器等模块。它们各自的功能分别是:①客户端发送 XQuery 查询请求,XQJ 接口接收查询请求,同时将请求转发给语法分析模块检查查询有效性;②集成规则负责处理不同数据源的数据集成规则,并将它交给 XQuery 引擎;③XQuery 引擎根据集成规测通过适配器查询不同的数据源;④不同的适配器负责访问不同的数据源,如 ODF 适配器负责从 ODF 数据源获取数据,并将查询结果以 XML 数据返回;⑤结果处理主要根据查询得到的 XML 结果进行处理,最后将结果返回给客户端。

数据源层包含 XML,ODF 和 OOXML,或者 UOF 等文件。这些文件来源于不同的应用系统。本文关注的

是针对 ODF 格式的文档的数据查询与集成,关于不同应用系统如何支持和转换 ODF 文档,这里不作过多详述。微软的文件格式转换器资料可以参阅参考文献[4]。

本文提出这个的基于 XQuery 的针对 ODF 系统数据集成框架,同样也适用微软的 OOXML 标准文档,或者我们中国提出的自己的文档标准 UOF。只要根据不同的文档标准设计相应的适配器即可。这里我们只是把重点放在 ODF 文档的数据集成中来。其它两种文档标准可以参阅参考资料[5]。

4 数据集成框架实现

上面提出的数据集成框架的实现过程主要有两个方面的关键技术需要解决,即 ODF 适配器的设计、XQuery 引擎的实现。

4.1 ODF 适配器的设计

ODF 适配器主要负责 ODF 文档的访问和查询。我们可以使用 XQuery 访问和查询 XML 文件,但是如何对 ODF 文件处理? ODF 文件是打包的文件,并且经过了 zip 格式的压缩。对于 JAVA 实现的 XQuery 引擎,可以使用 XQuery 标准函数 fn:doc 处理。

代码段 1:

```
doc ("jar:file:///C:/Books.ods!/content.xml");
```

表 3 是图书清单的 ODF 文件,它包含图书的名称,作者,出版日期和价格等信息。我们可以使用代码 2 查询所有价格低于 5 的图书清单。

表 3 Excel 转换的 ODF 文件 books.ods 中的数据

TITLE	AUTHOR	PUB – DATE	PRICE
Pride and Prejudice	Jane Austen	2002 – 12 – 31	4.95
Wuthering Heights	Charlotte Bront	2002 – 12 – 31	6.58
Tess of the d'Urbervilles	Thomas Hardy	1984 – 5 – 1	4.95
Jude the Obscure	Thomas Hardy	1998 – 9 – 1	4.95
The Big Over Easy	Jasper Fforde	2005 – 7 – 11	16.47

代码段 2:

```
declare namespace table =
"urn:oasis:names:tc:opendocument:xmlns:table:1.0";
declare variable $excel_doc :=
  doc("jar:file:///C:/Books.ods!/content.xml");
for $table in $excel_doc//table:table
  return
  for $row in $table/table:table-row[position() > 1][table:table-cell[4]<5]
    let $title := $row/table:table-cell[1]
    let $price := $row/table:table-cell[4]
  return
    <result>
     <title>{$title//text()}</title>
     <price>{$price//text()}</price>
    </result>
```

4.2 XQuery 引擎

准备好数据源和 XQuery 之后,我们可以使用 XQuery 的一个 JAVA 实现—SAXON 来完成 XQuery 的执行。Saxon 是由英国的 Michael Kay 开发的 XSLT 与 XQuery 处理器。从 2003 年中发布的 7.6 版开始,Saxon 增加了对 XQuery 的支持。Saxon 中的 XQuery 支持本质上由一个 XQuery 解析器(也是 XPath 解析器的扩展)组成;解析器与 XSLT 处理器一样,生成相同的内部可解释的代码。Saxon 现有两个版本。Saxon – B 8.1 是一个无模式的处理器,并且作为开源产品免费在 SourceForge(http://saxon.sf.net/)上提供。

Saxon 8.8 实现了 XQJ,我们可以使用 Saxon 的 XQJ API 在 JAVA 应用程序中调用 XQuery。Saxon 自己定义了 XQJ 接口,打包在 saxon8 – xqj.jar,并且 XQJ 接口从规范中的包名 javax.xml.xquery 改为 net.sf.saxon.javax.xml.xquery。这是为了避免和其他版本的 XQJ 冲突,毕竟 XQJ 规范还是个草案。

代码段 3 简单示例如何通过 XQJ 执行一个简单的 XQuery 查询。

代码段 3:

```
XQDataSource ds = new SaxonXQDataSource();
XQConnection conn = ds.getConnection();
XQPreparedExpression exp = conn.prepareExpression("<a b='c'>{5+3}</a>");
XQResultSequence result = exp.executeQuery();
```

```
while (result. next()) {
System. out. println(result. getItemAsString());
}
```

在我们的数据集成框架下,使用 saxonb8 – 9 – 0 – 4j 作为 XQuery 引擎来执行我们的 XQuery 代码。

5 总结

随着对 ODF 标准的日益推广,越来越多的文档尤其是办公文档会采用 ODF 格式,那么对于 ODF 文件的数据集成变得越来越重要。众多的应用系统支持统一的标准,也带来了数据集成的便捷性。本文提出了一个采用 XQuery 语言,对 ODF 文档的数据集成的模型,并对关键的技术进行了研究。这种方案具有一定的扩展性,只要开发相应的适配器就可应用于其它类似的标准,如微软的 OOXML 和我们中国自己提出的 UOF 文件标准。由于 ODF 文档的复杂性,要完成成熟的数据集成框架,今后还有很多工作要作,比如 ODF 文档格式作进一步的解析,查询效率等问题。

参考文献

1 OpenDocument v1. 1 Specification. 1 Feb 2007. http://docs. oasis – open. org/office/v1. 1/OS/OpenDocument – v1. 1 – html/OpenDocument – v1. 1. html.

2 XQuery 1.0: An XML Query Language. W3C Recommendation 23 January 2007. http://www. w3. org/TR/xquery/.

3 JSR 225: XQuery API for Java TM (XQJ). 11 Jun, 2007. http://www. jcp. org/en/jsr/detail? id = 225.

4 OpenXML/ODF Translator Add – in for Office . http://sourceforge. net/projects/odf – converter.

5 Standard ECMA – 376 Office Open XML File Formats. December 2006. http://www. ecma – international. org/publications/standards/Ecma – 376. htm.