XML 数据文档及其处理技术探讨*)

The Study of XML Data Document and it's Processing Technology

夏小彬　严小卫

(广西师范大学计算机系　桂林 541004)

Abstract　Two programming models to process XML document, i. e. DOM and SAX, are described in details. A new idea of combining them is proposed.

Keywords　HTML, XML, SAX, DOM, API

1. 引言

目前 Internet 上描述网页信息的 HTML 语言的元素类型是通用和描述性的，既不具备可扩展性，也不能有效地表示信息的结构和意义。这也就是许多搜索引擎往往针对用户输入的关键词却返回大量垃圾数据的根源。XML 的出现给了人们一条解决这一问题的途径。XML 是用于描述结构化数据的元标记语言，是结构化文档和数据的统一格式，提供了一个对数据的内容进行更精确声明，及为对多个松散的应用进行更有意义的搜索，得到精准的结果集提供了一个标准。

由于 XML 所具有的一系列独特的优点使它迅速风靡全世界，并成为 Internet 上的世界语，于是对各种来源的 XML 文档的处理也就成为人们关注的热点。本文详细论述了两种对 XML 文档进行处理的编程模型：DOM、SAX，并针对二者的长处和不足提出了结合使用 DOM、SAX 的新设想。

2　XML 文档的处理

为了正确处理 XML 文档，必须了解 XML 文档的处理流程。在处理过程中 XML 文档是作为文本流从源位置流向目标位置的，针对 XML 文档处理的这一特点，目前有两种解析器模型用于处理流向应用程序的 XML 文档流：SAX 模型和 DOM 模型，两种模型都定义了一套对 XML 文档进行操作的 API。一般用户对 XML 文档的处理都是借助某个 XML 解析器进行，目前可用的 XML 解析器有 SUN 的 JAXP、IBM 的 XML4J、XML4C，ORACLE 的 EXPAT 和 Microsoft 的 MSXML，其中 Microsoft 在今年 9 月发布的 MSXML3.0 提供了丰富的功能，包括文档合法性验证，对 XSLT/XPath 的完全支持，对 DOM 和 SAX 服务接口的实现，用 XPath 在 DOM 中查询时对名字空间的支持，能通过 HTTPRequest 进行远程调用请求，并支持服务器端的安全连接等，它是一个优秀的 XML 解析器。在本文的例子中将使用 MSXML3.0 解析器演示处理下面的 XML 文档。

一个完整的 XML 文档示例

```
<? xml version="1.0"?>
<books>
  <book id="TP21">
    <isbn>7-5635-0422-2</isbn>
    <title>XML 理论和应用基础</title>
    <author>孙一中</author>
    <price>36.00</price>
    <publisher>北京邮电大学出版社</publisher>
  </book>
  <book id="TP27">
    <isbn>7-302-03956-9</isbn>
    <title>基于 C++CORBA 高级编程</title>
    <author>徐金梧</author>
    <price>80.00</price>
    <publisher>清华大学出版社</publisher>
  </book>
</books>
```

下面讨论 SAX 和 DOM 两类编程模型的特点，及如何利用 MSXML3.0 对该文档进行处理。

2.1　SAX

SAX 解析器是基于事件触发处理的，SAX 解析器顺序读过 XML 文档并基于它在文档中遇到的特定标记产生相应的事件。与人们了解的基于事件的编程有一点不一样，在一般基于事件的编程中，用户可以定制事件，然而利用 SAX API 进行基于事件的编程时，事件只能由解析器产生，如考虑下面的文档段：

```
<book>
    <author>孙一中</author>
</book>
```

当 SAX 解析器处理该文档时，它顺序查找文本流中的元素开始标记和结束标记，并触发一系列相应的事件，参见图 1。

*)基金项目：广西自然科学基金(0007008)和广西十百千人才工程专项资金资助。

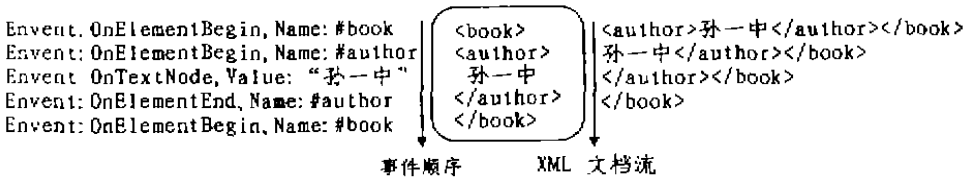

图 1 SAX 解析器工作示意图

标记的名字传送到事件处理器，并附加上该元素包含的属性元素的一个数组，然后程序应用 SAX 解析器根据标记的内容来解释该标记。SAX 解析器是解析 XML 文档的第一类解析器。该类解析器在处理拥有上百万节点的大 XML 文档时非常有效。然而由于它在解析过程中没有保存文档的上下文信息，当它碰到一个新标记时根本就不知道有关上一个标记的任何信息，它把处理所有这些相关信息细节的负担转嫁给了编程人员。因此，SAX 编程接口是一种较底层的接口。事实上可以把 SAX 解析器看作是一种推类型的解析器，SAX 产生一系列的事件，用户可以选择感兴趣的事件，并为之提供包含处理方法的事件处理器，对于没有提供事件处理器的方法，SAX 解析器只简单地忽略它。如下的程序代码提供了一个事件处理器，它包含两个方法：IVBSAXContentHandler_endElement 和 IVBSAXContentHandler_characters，程序执行的结果是在文本框中显示 Book1：XML 理论和应用基础。

SAX 模型编程的示例代码

```
'程序运行的启动代码
Dim reader As New SAXXMLReader
Dim contentHandler As New ContentHandlerImpl
Set reader.contentHandler = contentHandler
reader parseURl (Text1.text)

'程序实现的事件处理器，其中提供了对标记结束事件和字符数据事件的处理方法实现
Implements IVBSAXContentHandler
Dim ResultTitle As String
Dim BookNo As Integer
Private Sub IVBSAXContentHandler_endElement(strNamespaceURI As String, strlLocalName As String, strQName As String)
  If strLocalName = "title"Then
    If InStr(1,ResultTitle,"XML",vbTextCompare)Then
      BookNo = BookNo + 1
      Form1.TextBox1 text = Form1.TextBoxl.text & _
      "Book" & bookNo & ":" & ResultTitle & vbCrLf
    End If
  End If
End Sub
Private Sub IVBSAXContentHandler_characters ( text As String)
    ResultTitle = text
End Sub
```

2.2 DOM

DOM 解析器是基于文档对象模型的，当 DOM 解析器在读取 XML 文档时，把它分解成一个个独立的对象，如元素、属性、注释等，然后在内存中创建出相应的树结构，程序对文档内容的任何操作必须在这棵树完全构造完成后才能有效，DOM 解析器也提供了供用户查询解析是否完成等状态信息查询功能。一旦这棵树构造完成，就可以很容易独立地引用和操作每一个对象或节点，从任一元素的处理跳到其他任一元素去继续处理；能够容易地查找到当前节点的父节点、祖父节点、兄弟节点和孩子节点；还可以利用 XPath 执行跨节点的移动，从而构造出一棵新的更易于使用的文档结构树。这些功能对于执行文档内容修改、保存和进行复杂查询都是非常关键的。然而对一个大文档，DOM 的处理消耗大量的内存，DOM 解析器实际上工作于 SAX 解析器的顶层。前面的文档经 MSXML3.0 用 DOM 模式解析后得如图2的文档树。

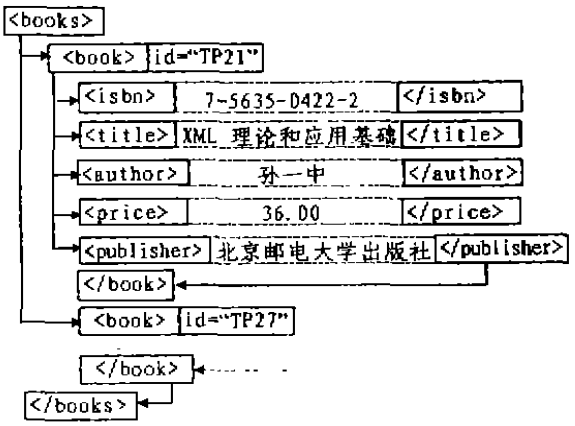

图2 用 DOM 模型解析后得到的文档结构树

下面是利用 DOM 模型的 API 实现如上述程序同样功能的程序代码。

DOM 模型编程的示例代码

```
Dim xmlDoc As New MSXML2.DOMDocument30
xmlDoc. Load ("C:\Inetpub\wwwroot\xmlweb2\books.xml")
Dim targetNodeList As IXMLDOMNodeList
Set targetNodeList = xmlDoc.selectNodes("//title")
Dim BookNo As Integer
Dim nodesize As Integer
Dim targetNode As IXMLDOMNode
For nodesize = 1 To targetNodeList length
    Set targetNode = targetNodeList nextNode
    If InStr(1,targetNode Text,"XML",vbTextCompare)
    Then
```

```
BookNo=BookNo+1
Form1. Text1. Text = Form1. Text1. Text & "Book" &
    BookNo & ":" & targetNode. Text & vbCrLf
End If
Next nodesize
```

2.3 SAX 和 DOM 的优缺点对比

	SAX	DOM
优点	能轻松处理拥有上百万个节点的大文档，且内存的消耗不会随文档增大而增多，能显著节约内存和提升处理效率	能执行强大的 XSLT 变换，用多种不同类型的风格样式单为同一个文档创建出多个视图
	能随时根据找到信息的情况终止文档解析，从而避免了无谓的遍历文档	DOM 模型的树形结构将自动维护包含文档上下文信息的复杂数据结构，能执行复杂的 Xpath 过滤功能
	SAX 可用高层的对象，如股票代号，新闻等从几个大文档中非常简单快速地创建出新文档，采用 DOM 则只能利用底层的元素，属性和处理指令来构造新文档；	在内存中建立文档的树形结构，能随机访问数据，并能随意修改文档内容并保存修改
	SAX 是属于底层的 API，给程序员更多的控制权	DOM 属于高层的 API，编程方便，得到更广泛的支持
缺点	不能保存文档的上下文信息；不能随机访问文档结构；不能实现复杂的查询；目前缺乏浏览器的支持	不能适应大文档的处理；占用系统资源多；处理速度不如 SAX 快；不能终止解析过程

3 DOM 和 SAX 的结合使用的设想

我们可以利用给 SAX 体系结构添加索引信息来绕过 DOM 模型的尺寸限制，这对于那些内容并不经常更新的大 XML 文档进行只读操作来说是很好的选择。具体做法在 SAX 解析器顺序读 XML 文档时利用一个索引程序定位每一个元素的开始和结束位置来建立一个有关元素位置的索引表。当执行查询时，标记名在 XML 文档中的位置是通过文件定位指令（如 Seek）在标记位置索引表中检索来得到的，由于 XML 文档的结构完整性，所以一旦给定的元素及其内容被检索出来，那么在查询中的子元素也肯定在检索出来的信息的子集中，通过选择性的抽取元素，就能够最终检索到任何需要节点，而且还可以将检索到的节点位置信息添加到节点位置索引表中以便于后继再次查询时提高效率。

参 考 文 献

1 孙一中，等. XML 理论和应用基础. 北京：北京邮电大学出版社，2000
2 Goldfarb C F，等. XML 实用技术. 北京：清华大学出版社，1999
3 Available at：http：// www. microsoft. com/xml/
4 Available at：http：// www. w3. org/TR/REC-xml
5 Available at：http：// www. megginson. com/SAX

（上接第52页）

以 Dmax＝150ms，在速率为10Mbit/s 的以太网上，传输一个多媒体单元（一帧视频图像或一个音频片段，本例中其最大长度设为4kB）的最小延迟时间 Dmin 为：

Dmin＝一个多媒体单元的大小/网络的最大传输速率

Dmin＝4kB/10Mbit/s＝3.2ms

在本例中，音频片段的采集间隔 θm＝1000ms，视频帧速为15帧/秒，因此 θs＝66.7ms。

音频需要预设的最小和最大的缓冲单元个数分别为：

$$ABmin=\lceil(150ms-3.2ms)/1000ms\rceil=1$$

$$ABmax=\lceil 2\times(150ms-3.2ms)/1000ms\rceil=1$$

视频需要预设的最小和最大的缓冲单元个数分别为：

$$VBmin=MAX\{\lceil(150ms-3.2ms)/66.7ms\rceil,\ \lfloor(1-1)\times 1000ms/66.7ms+1\rfloor\}=3$$

$$VBmax=MAX\{\lfloor 2\times(150ms-3.2ms)/66.7ms\rfloor,\ \lfloor((1-1)\times 1000ms+150ms-3.2ms)/66.7ms+1\rfloor\}=5$$

根据计算结果，我们将客户端音频的接收缓冲单元数设为1，而视频的接收缓冲单元数设为5。实验结果表明，系统能够很好地实现多媒体的实时连续性通信并且音视频达到很好的同步，满足 QCIF 视频的表现质量要求。

小结 在分布式多媒体通信系统里，适当地为同步源设置多媒体对象缓冲区，是保证多媒体通信的实时性和连续性，以及实施流间媒体同步控制的必要手段之一，本文对同步时所需要的有界缓冲区大小的计算结果，尽管是在较为理想的通信条件下得出，但它仍然为实际的多媒体通信系统设计提供了可供参考的同步参数之间的量化依据。实际应用表明，运用本文的参数计算结果，能够很好地利用系统资源，并有效地保证了多媒体通信中的媒体表现质量要求。

参 考 文 献

1 胡晓峰，李国辉. 多媒体系统. 人民邮电出版社，1997
2 郑庆华，等. 分布式多媒体同步中表现质量的参数计算. 通信学报，1999，10(20)
3 张明德，王永东. 视频会议系统原理与应用. 北京希望电子出版社，1999
4 魏铁军，陈俊亮. 基于漏窗机制实时自适应多媒体同步. 通信学报，1999，5(20)
5 李子辉. 基于 IP 组播的实时媒体传输研究与实现：[学位论文]. 广州：华南师范大学，2000