

XPath 技术的解析

段晓娟 徐长梅 孙宁 (长沙大学计算机系 410003)

```
Set pnts = Nothing
objGCS.EndCollection
Set fgeoCreateRectangle = objGCS.Geometry
Set objGCS = Nothing
Set objpnt = Nothing
Exit Function
```

ErrorHandler:

```
MsgBox Err.Description, vbCritical + vbOKOnly, "提示"
```

End Function

4 Geomedia 扩展类的使用

Geomedia 扩展类的使用极为简单, 可以把它像普通的类 (如 Microsoft 的 ADO 类) 一样使用, 使用该扩展类的基本步骤如下:

(1) 首先是在 VB 集成开发环境中, 通过菜单 Project \ Reference... 实现对扩展类 ClsMapAccessEx 的引用。

(2) 在客户程序中定义一个该扩展类的全局对象变量, 并将其实例化 (即 Dim MapEx as New ClsMapAccessEx)。

(3) 通过相关函数 (ConnectGDatabase) 建立客户程序与空间数据库的连接并完成初始化, 如果成功建立连接, 则可以执行和使用扩展类的其他函数和方法。

(4) 扩展函数的调用。如创建一个矩形对象, 其代码如下:

```
set aOBJ=MapEx.CreateRectangle(0,0,0,10,20)
```

(5) 在终止客户程序之前, 应先断开客户程序与空间数据库的连接, 并将该扩展类的全局实例变量设置为 Nothing, 以释放空间。

5 结语

本文构建的扩展类共包括近 80 个扩展函数, 这些函数几乎涵盖了 Geomedia 中用于图形访问的全部功能。该扩展类的构建, 规范并统一了与图形访问相关的编码, 使该部分编码具有更高的可靠性, 最大限度地实现了代码的共享, 同时也使代码的维护和应用系统的集成过程变得更为简单可靠。该扩展类是笔者在开发某市电信资源管理系统时构建的, 实际应用证明了该扩展类的使用, 能有效保证所开发软件的质量, 并能显著提高软件开发的效率。 ■

参考文献

- 1 张霞, 崔逊学, 应用 GeoMedia 开发嵌入式 GIS 的方法 [J], 计算机系统应用, 2002, (3): 22-24.
- 2 龚健雅, 当代 GIS 的若干理论与技术 [M], 武汉测绘科技大学出版社, 1999.

摘要: 本文详细讨论了 XPath 的工作原理、数据模型、定位路径等主要语法, 并通过示例说明其具体使用方法。

关键词: XPath XML 数据模型

1 XPath 数据模型

XPath (XML Path Language) 是一种对 XML 文档的内容进行定位、检索的语言, 是后续更强大的数据检索语言如 XQuery 的基础。XPath 因使用路径标记在 XML 文档的层次结构中进行导航而得名, 其工作方式与语法有些类似操作系统中用于文件定位的路径以及互联网中用于资源定位的 URL, 当然, XPath 比后者复杂得多。XPath 不独立使用, 主要嵌入在 XSLT、XPointer、DOM 等宿主语言中应用, 比如, 在 XSLT 的应用中, XPath 用在模板 (template) 中来检索数据以及定义匹配模式 (pattern)。

数据模型即 XPath 看待它操作的 XML 文档的方式, 了解数据模型有助于更好地了解 XPath 的语法以及工作方式。XPath 将 XML 文档视为一个节点树, 如图 1 所示。节点有七类, 包括根节点、元素节点、属性节点、文本节点、命名空间节点、处理指令节点、注释节点。XPath 对每种节点都定义了计算其字符串值的方法, 不同的节点计算方法可能不同。

文档顺序 (document order) 定义了每个节点在 XML 文档中出现的顺序, 此顺序与 XML 文档中节点的物理位置 (开始标签) 一致, 如图 1 左。如此, 根节点为文档的第一个节点, 每个元素节点出现在其子节点之前等。

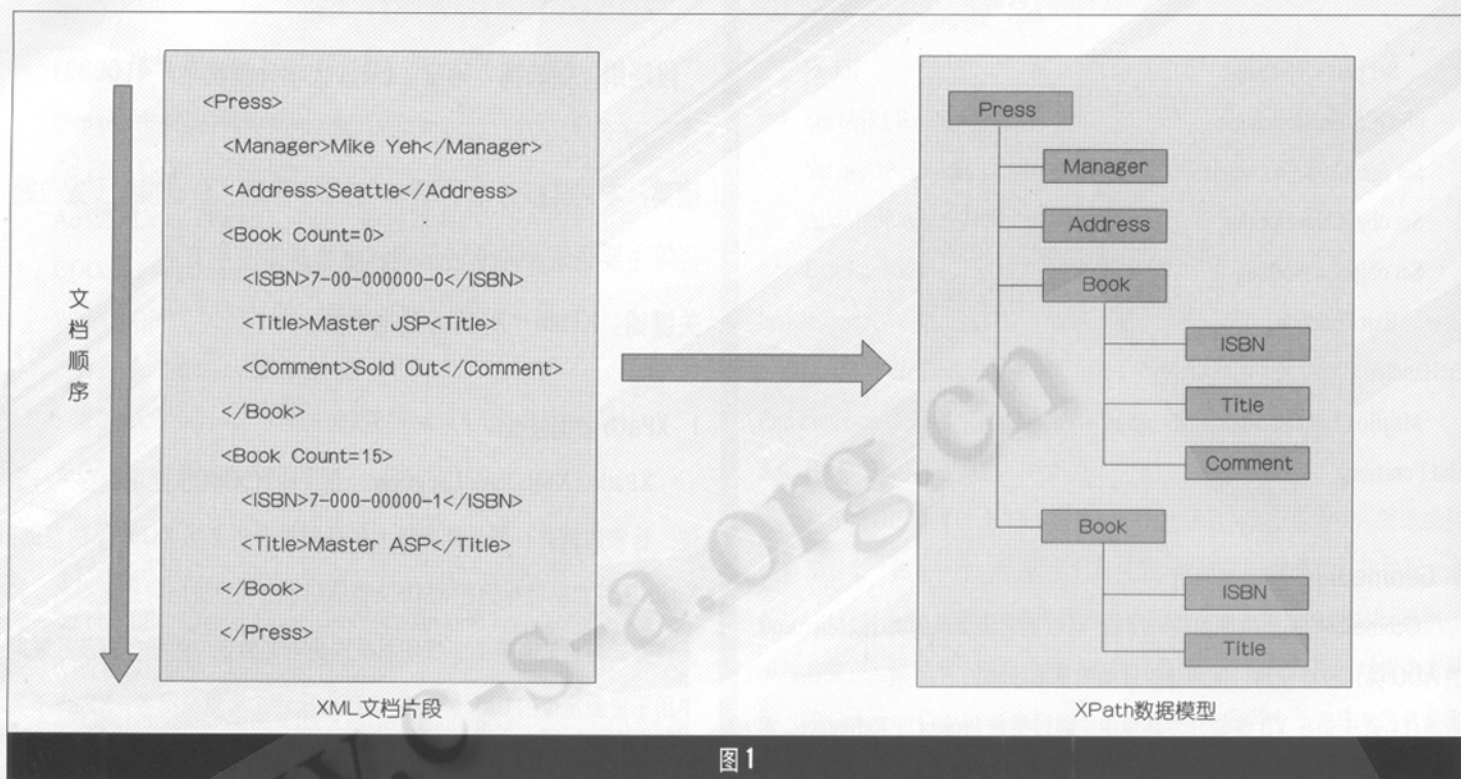
根节点和元素节点可以有有一个有序列表的子节点 (即子节点之间有一定的顺序关系, 不能任意排列)。节点不共享子节点, 即除了根节点, 每个节点都有唯一的、明确的父节点, 父节点可能是根节点或元素节点。一个节点的后代包括它的子节点以及子节点的后代, 如图 1 右。值得注意的是, 属性节点肯定有其父节点, 但属性节点不是其父节点的子节点, 即属性节点不被视为子节点。

2 语法

2.1 表达式 (expression)

XPath 主要的语法结构是表达式, 它计算的结果是一个对象, 类型为

Analysis of XPath Technology



四种基本类型之任何一种：节点集、布尔值、数字、字符串。

在计算表达式时，需要考虑所谓的上下文(context)，上下文由宿主语言决定(可参考相应的宿主语言的说明)，包括以下几个部分：一个节点，称为上下文节点(context node)；一对非零正整数，称为上下文位置(context position)和上下文大小(context size)；一套变量绑定；一个函数库；一套表达式范围内的命名空间声明。

2.2 定位路径 (location path)

定位路径是XPath最重要的一种表达式，它选择与上下文节点相关的一个节点集 (node-set) 作为其计算结果。XPath有两种定位路径：相对路径和绝对路径。在相对路径中，每一个定位步骤都选择与上下文节点相关的一套节点，然后在这一套节点中的每一个节点都作为后续定位步骤的上下文节点进行选择，如此从左到右进行选择，最后选择的所有节点组合在一起，得到定位路径的结果。

2.3 定位步骤 (Location Steps)

定位步骤有三个部分：

- 一个轴：指明定位步骤选择的节点与上下文节点在节点树上的关系；
- 一个节点测试：指明定位步骤选择的节点的类型和扩展名称；
- 零个或多个断言：使用任意的表达式进一步精简选择的节点集。

一个定位步骤选择节点的过程为：初始的节点集由那些与上下文节点有相应关系的节点构成，其中关系由轴指出，同时，这些节点还必须有着由节点测试部分指定的节点类型和扩展名称，最后，初始的节点集将由断言逐个过滤，以得到此定位步骤选择的最终结果。

任何一个轴要么是前向轴，要么是后向轴。包括上下文节点以及其后续节点(按照文档顺序)的轴称为前向轴；包括上下文节点以及其前驱节点(按照文档顺序)的轴称为后向轴。对self轴来说，它最多包括上下文节点一个节点，称其为前向轴或后向轴都可以。

需要注意的是，每个轴都有其首选节点类

型，即在进行节点测试时，节点的类型首先应该符合轴的首选节点类型。对attribute轴，首选节点类型为属性节点；对namespace轴，首选节点类型为命名空间节点；对其他轴，首选节点类型为元素节点。

断言对节点集进行进一步过滤，节点集中的每一个节点都对断言中的表达式进行计算，如果计算结果为true，此节点包括在结果节点集中，否则被过滤掉。断言一般针对节点的上下文位置、上下文大小、节点名称、属性值等进行计算。其中，上下文大小为原节点集节点的个数，上下文位置为当前计算节点在原节点集中的位置，从1开始计数，上下文位置根据轴的方向计算方法略有区别。

3 示例

下面通过一些例子来说明XPath的使用方法。采用一个工具软件XMLSpy来辅助实验，XMLSpy为业界领先的XML编辑工具，针对XML相关文档提供强大的编辑功能。当然，在

[Applied Technique]

此主要是利用 XMLSpy4.3 中新增加的一项称为“Evaluate XPath”的功能，通过直接输入 XPath 表达式来计算表达式的结果，完全符合实验的要求。以下为 XMLSpy 在运行“Evaluate XPath”功能时的界面，在有节点被选中时，通过单击选中的节点在 XML 文档中定位此节点。

首先，我们在 XMLSpy 中新建一个 XML 文档，输入以下 XML 片断，然后将光标定位在第二个 Book 节点的开始标签中，对 XMLSpy 而言，即为选中了第二个 Book 节点。我们将使用 XPath 对其他的节点进行定位实验，实例使用简化语法，并注意我们在“Evaluate XPath”对话框中选择“XPath Origin”为“From Selected Element”，即以当前节点为 XPath 表达式的参考原点：

```
<Press>
<Book Count="30">
  <ISBN>7-00-000000-0</ISBN>
  <Title>Master JSP</Title>
</Book>
<Book Count="10">
  <ISBN>7-000-00000-1</ISBN>
  <Title>Master ASP</Title>
</Book>
<Book Count="40">
  <ISBN>7-0000-0000-2</ISBN>
```

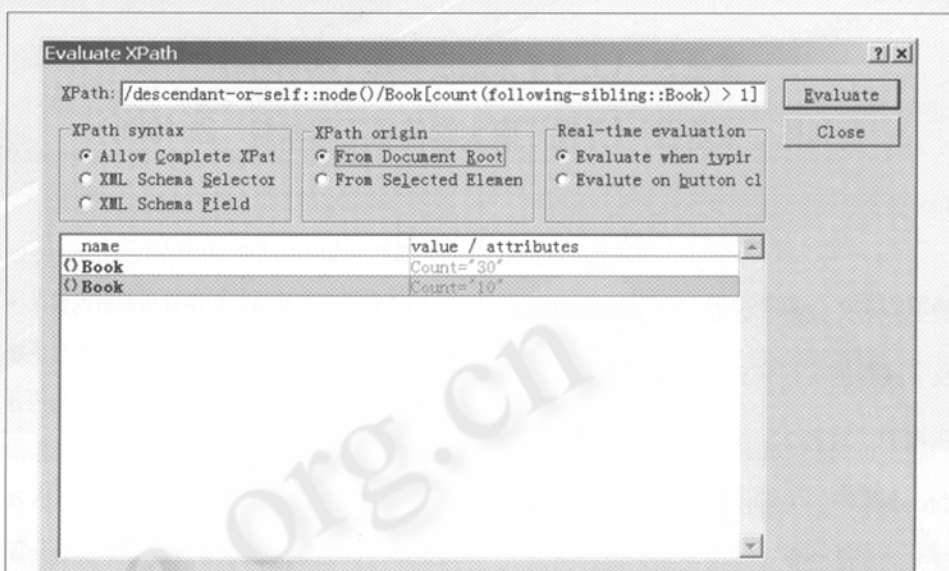


图 2

```
<Title>Master Java</Title>
</Book>
<Book Count="25">
  <ISBN>7-00000-000-3</ISBN>
  <Title>Master XML</Title>
</Book>
</Press>
```

下表对 XPath 表达式进行解释（从左到右）
经过分析看出，整个 XPath 表达式的结果是：选择那些后续兄弟节点个数大于 1 的 Book 节点，即为文档中的第一个和第二个 Book 节点，第三个 Book 节点后续兄弟节点个数为 1，第四个 Book 节点后续兄弟节点个数为 0，二者均被过滤。

4 结束语

XPath 目前功能还不完善，比如没有关联查询，不能对数据进行排序、汇总、统计、分组，过滤机制也较弱，这些功能有的可以由宿主语言实现，有的则需要在 XPath 后续版本中解决。■

参考文献

- 1 XPath 1.0 规范 (<http://www.w3.org/TR/1999/REC-xpath-19991116>).
- 2 XML 1.0 规范 (<http://www.w3.org/TR/2000/REC-xml-20001006>).
- 3 XML InfoSet (<http://www.w3.org/TR/2001/REC-xml-infoset-20011024>).

表 1

XPath 表达式	说明
/	选择 XML 文档的根节点（类似文件系统中的根目录）
descendant-or-self::node()	选择当前节点及所有后代节点，形成一个节点集。在本例中，当前节点为文档根节点，因此，此表达式选择了文档中所有节点。
Book	选择节点集中的 Book 节点。在本例中，即为所有的 4 个 Book 节点
[< 表达式 >]	表示一个断言。中间的内容为一个表达式。断言对节点集中每个节点都对表达式进行计算，结果不为 true 的节点被过滤
count()	统计函数。
following-sibling::Book	选择当前节点的后续兄弟节点中的 Book 节点
count(following-sibling::Book)>1	布尔表达式。对上一表达式中的 Book 节点进行统计，如统计结果大于 1，则返回 true，否则返回 false