

●谈春梅 段卫华 刘 伟

电子信息资源数据库检索系统的开发与实现^{*}

摘 要 采用 XML 查询技术和显示格式,可开发出电子信息资源数据库检索系统,能实现模板基本查询、格式化显示及 XML 文档高级查询等功能。其乱码输出、模糊匹配等问题也可解决。参考文献 5。

关键词 电子信息资源数据库 XML 技术 网络检索 开发设计

分类号 G250.74

ABSTRACT By using XML query processing technology, we can develop systems for the access to electronic information resource databases, and actualize template basic search, formatted display, XML file advanced search and other functions. The strange character output, fuzzy matching and other problems can also be solved. 5 refs.

KEY WORDS Electronic information resource database. XML technology. Network search. Development and design.

CLASS NUMBER G250.74

当前,电子信息资源组织的一种有效方式——电子信息资源数据库,已成为图书馆界开发研究的重要课题。我们借助于图书馆计算机网络平台,进行了电子信息资源数据库系统的开发设计。

电子信息资源数据库系统由 MARC 数据收集子系统、MARC 数据与大型关系型数据库相互转换子系统和电子信息资源数据库检索子系统 3 大模块组成。其中电子信息资源数据库检索子系统是电子信息资源数据库系统的重要组成部分。本文将着重介绍该子系统的主要开发技术及其程序的设计与实现。

源数据库系统的重要组成部分。本文将着重介绍该子系统的主要开发技术及其程序的设计与实现。

1 系统的功能模块及主要开发技术

电子信息资源数据库检索子系统主要包括表单提问、网络基本查询、网络高级查询和查询结果数据显示等功能。

每个功能模块既相互独立又相辅相成。这些功

(上接第 50 页)

- 5 McCahill, M. P. & Erickson, T. Design for a 3D spatial user interface for Internet Gopher. Proceedings of ED-MEDIA 95, 39-44. Charlottesville, VA: AACE
- 6 Andrews, K. Visualising cyberspace: Information visualization in the Harmony Internet browser. Proceedings of Information Visualization, 97-104. 1995. Los Alamitos, CA: IEEE
- 7 Hendley, R. J., Drew, N. S., Wood, A. M., & Beale, R. Narcissus: Visualizing information. Proceedings of Information Visualization, 90-97. 1995. Los Alamitos, CA: IEEE
- 8 Mukherjea, S. Foley, J. D., & Hudson, S. Visualizing complex hypermedia networks through multiple hierarchical views. Proceedings of CHI'95, 331-337. New York: ACM
- 9 Benford, S., Snowden, S., Greenhalgh, C., Ingram, R., Knox, I., & Brown, C. VR-VIBE: A virtual environment for cooperative information retrieval. Eurographics'95, 349-360

- 10 Hennje, M., Kunkel, C., & Willett, A. LyberWorld: A visualization user interface supporting fulltext retrieval. Proceedings of ACM SIGIR, 249-259. 1994. New York: ACM
- 11 Gershon, N. D., LeVesseur, J., Winstead, J., Croall, J., Pernick, A., & Ruh, W. Visualizing Internet resources. Proceedings of Information Visualization, 122-128. 1995. Los Alamitos, CA: IEEE
- 12 Marti A. Hearst. Tilebars: Visualization of term distribution information in full text information access. In Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems, Denver, CO, May 1995. ACM

周 宁 武汉大学教授,博士生导师。通讯地址:武汉大学信息资源中心。邮编 430072。

文燕平 武汉大学信息管理学院 2001 级博士研究生。通讯地址:武汉。邮编 430072。(来稿时间:2002-04-09)

^{*}本文系江苏省哲学社会科学规划基金项目(省 05)研究成果。

能模块已基本实现该系统设计的技术要求,并能正常运行。

主要开发技术,是以新型的 SQL Server 2000 为电子信息资源后台数据库(电子信息资源 MARC 数据库被转换为 SQL Server 2000 关系型数据库),该关系型数据库为 XML 数据查询提供了支持,从而大大方便了跨平台、跨系统间异构数据的交换与存取。

采用先进的 XML (eXtensible Markup Language,可扩展标注语言)技术实现了电子信息资源数据库的基本查询和高级查询。与 ASP (Active Server Pages,活动服务器主页)等查询技术相比,XML 技术不仅易于编程和实现,而且对服务器的要求比较低,服务器端的负担得以减轻,网络效率也随之提高。同时,使用 XML 显示格式,使得用户通过浏览器得到的不是传统的 HTML 文档,而是 XML 文档,从而克服了 HTML 文档内容与显示格式不能分离的局限性。

2 基于 XML 技术的电子信息资源数据库检索程序的设计与实现

SQL Server 2000 数据库管理系统提供的 XML 支持有 3 种方法,即 URL 查询、模板查询及指定在带批注的 XDR 架构上执行的 XPath 查询。电子信息资源数据库基本查询采用了 XML 模板查询。

2.1 XML 模板基本查询

模板查询可直接在 URL 中指定模板文件,模板文件允许加入查询参数,为查询程序设计带来很大的方便。模板查询主要采用了 SQL 语句和 XSLT XPath 查询语言。XSLT 是一种转换工具,适用于对查询结果进行处理和重建。同时,XSLT 还可利用所提供的许多函数来实现查询功能。XPath 是 XML 分级命令处理工具,非常适用于访问 XML 文档的独立节点。使用 XML 模板查询方法实现了电子信息资源数据库的基本查询(主要源代码见附录 1),其程序设计要点是:

(1)指明 XML 的版本号及编码语言(版本号为 1.0,编码语言为 GB2312),并指明采用模板查询方法对数据库进行查询。

(2)定义具有检索意义的参数。以 SQL Server 2000 为电子信息资源后台数据库,分别有中文电子期刊数据库 ZWDZQK、外文电子期刊数据库 WWDZQK 和电子论坛数据库 DZLT 等。定义的参数,以 ZWDZQK 数据库为例,分别对应题名(TITLE)、责任者(SPONSOR)、出版者(PUBLISHER)、主题词(SUBJECTWORDS)、关键词(KEYWORDS)、国际标准刊号

(ISSN)、分类号(CLASSNUMBER)、内容描述(DESCRIPTION)等。

(3)使用 SQL 查询语句编写查询程序。从 ZWDZQK 数据库中查询符合上述所有参数的记录,并取出需要显示的所有字段(包括上述具有检索意义的字段以及用户可直接超链接的电子资源定位与检索的所有子字段)。

(4)使用 AUTO 模式将查询结果作为嵌套 XML 元素返回。

2.2 XML 文档的高级查询

采用 XML 模板查询方法实现了用户对电子信息资源的基本查询,为了进一步提高用户对电子信息资源的查全率和查准率,需要解决电子信息资源数据库的高级查询问题,该问题是网络查询程序设计的重要内容。对客户端浏览器上以模板查询方法得到的基本查询结果进行直接检索,不再经过服务器,原先集中于服务器端的大量运算负荷可以分布到客户端,从而大大降低了服务器的消耗,提高了网络效率,缩短了响应时间。

(1)采用 JavaScript 语言编写一个 HTML 文档程序格式化显示 XML 文档。

模板查询的结果为 XML 文档,要求该文档在客户端格式化输出。如果直接通过 SQL Server 系统本身进行处理,将会把 XML 文档转换成 HTML 文档,仍以 XML 的形式显示,但不保存原有的 XML 文档,这样,基于 XML 文档的高级查询就无法完成。因此,必须寻求新的方法来格式化显示 XML 文档。我们采用的方法是:用 JavaScript 语言编写一个 HTML 文档程序,在这个 HTML 文档中包含一个层 LAYER1,调用 Microsoft.XMLDOM,加载 XML 文档,然后利用 DOM 中的 transformNode 函数,用指定的 XSL 文档来格式化 XML 文档(主要源代码见附录 2)。其程序设计步骤是:

定义变量 xmldoc 和 xsl doc 为 Microsoft.XMLDOM 类型,并调用方法 Load,使 xsl doc 加载 ZWDZQK.xsl。这两个变量为全局变量,目的是使其他程序模块也可以调用这两个变量。

为变量 X 赋值。变量 X 值为模板所在的位置,并加入参数,其参数就是要查找的内容。然后 XMLDOC 加载变量 X,即加载数据库中所有符合查询条件的记录。程序中所描述的参数 y 是从 cookie 中读出的用户所要选择的查询方式,参数 z 是用户输入的查询内容。

将 XMLDOC 中内容以 XSLDOC 所规定的格式

输出到 LAYER1 中。其实质就是把数据库中所有符合要求的记录以 ZWDZQK.XML 中所规定的格式转换输出,替换 LAYER1 中的内容。这样,不仅实现了基本查询结果 XML 文档显示内容和显示格式的彻底分离,而且使在客户端进行高级查询开发设计成为可能。

(2) 使用 DOM 进行高级查询程序的开发与设计。

DOM (Document Object Model, 文档对象模型) 表示数据元素关系的方式非常类似于关系型数据库表示信息的方法,这使得利用 DOM 在数据库和 XML 文件之间移动信息变得相当简单。使用 DOM 进行高级查询程序开发设计,首先要解析 XML 文档,将 XML 文档分解为独立的元素、属性和注释,并以节点树的形式创建 XML 文档。DOM 将 XML 文档转化为可通过程序访问的节点树,这样,我们就可通过接口来访问 XML 文档,完成用户在客户端的高级查询(主要源代码见附录 3)。程序开发设计的具体步骤为:

定义变量。定义变量 searchxml,让这个变量从服务器端读取一个空的 XML 文档,这个文档仅仅规定了一个一级节点的名称为 root,一个二级节点的名称(如 ZWDZQK、WWDZQK)。这样做是为了使 searchxml 中能有和原来文档名称相同的根节点,其中变量 root 指向 searchxml 中的根节点。

定义具有检索意义字段的路径。我们仍然以 ZWDZQK 为例,如果参数 t (用户所选择查询的字段) 为 "title", j (将要查询的位置) 则为 "/ root/ zwdzqk/ title"; 如果参数 t 为 "sponsor", j 则为 "/ root/ zwdzqk/ sponsor"。

定义节点串变量 xmlsearchNodes, 查询 xml doc 中所有在查询路径上的二级节点。然后再定义节点串变量 xmlSearchChildNodes, 查询 xml doc 中所有在查询路径上的三级节点。

遍历 xmlSearchNodes, 如果与之相应的 xmlSearchChildNodes 的节点符合要求,则将这个节点及其子节点全部拷贝到 root 下,作为二级节点,也就是加入到新建的空 XML 文档中。

遍历结束后,这个文档中就是全部符合要求的节点。这时调用 xsl doc 来格式化显示这个文档,替换 LAYER1 中原来的内容。

3 几个难点问题的研究与解决

系统在开发设计过程中,遇到一系列难点问题,在课题组同志的共同努力下,一一得到了解决,其中重点解决了乱码输出、模糊匹配、参数传递等问题。

3.1 乱码输出问题

XML 模板查询输出的文档中有大量汉字,为了

使文档中的汉字能够正确输出,需要采用编码的方法加以抑制,我们使用了 GB2312 编码。而文档中汉字的输出仍为乱码。在 SQL Server 的 URL 查询中可以通过一个参数 OUTPUTENCODING 来指定输出编码。经尝试,这个参数同样可以应用于模板查询。从而抑制了乱码输出。

3.2 模糊匹配问题

在客户端为用户提供高级查询时,不仅要提供精确查询,而且更重要的是要提供模糊查询。因此,必须解决模糊匹配问题。由于大部分用户使用的是 IE5.0 及以下版本,而 IE5.0 及以下版本并不支持 XML 技术中专门用于处理字符串的一些新函数。我们在使用 XSLT XPath 的基础上,结合 DOM 编程来处理过滤 XML 文档。但是在 DOM 函数中,没有字符串包含函数。如果自己编写程序实现模糊匹配,效率势必会下降。我们利用了 indexOf 函数来确定字符串所在母字符串中的位置,如果这个值大于或等于零,那么子字符串就肯定在母字符串中。从而解决了模糊匹配的问题。

3.3 参数传递问题

当用户利用该系统检索所需电子信息资源时,要求按系统提供的检索界面填写表单,表单提交后需要得到显示结果,这就要求通过一个适当的程序将参数传递给模板,模板根据用户提供的信息进行查询,得到符合要求的记录,将查询结果传递给结果文档。仅仅通过 HTML 文档无法完成参数传递,一个常用的方法是用 ASP 编程将参数传递给下一个文档。但使用 ASP 编程,不仅相对速度比较慢,而且不利于用户二次开发。我们采用了系统的 cookie。主查询页面将这些参数写到系统的 cookie 中,然后再打开结果文档。结果文档在开始加载时运行程序,从 cookie 中读取参数,提交给模板查询,然后将模板查询的结果输出。

附录 1:

```
< ? xml version = '1.0' encoding = 'gb2312' ? >
< root
xmlns:sql = 'urn:schemas-microsoft-com:xml:sql' >
< sql:header >
< sql:param name = 'title' > </ sql:param >
< sql:param name = 'sponsor' > </ sql:param >
< sql:param name = 'publisher' > </ sql:param >
< sql:param name = 'subjectwords' > </ sql:param >
< sql:param name = 'keywords' > </ sql:param >
< sql:param name = 'ISSN' > </ sql:param >
< sql:param name = 'classnumber' > </ sql:param >
< sql:param name = 'description' > </ sql:param >
</ sql:header >
```

```

</ sql :query >
select * from zwdzqk
where Title like b/o '+' @title + b/o 'AND
Sponsor like b/o '+' @sponsor + b/o 'AND
Publisher like b/o '+' @publisher + b/o 'AND
Subjectwords like b/o '+' @subjectwords + b/o 'AND
Keywords like b/o '+' @keywords + b/o 'AND
Classnumber like b/o '+' @classnumber + b/o 'AND
Description like b/o '+' @description + b/o '
for xml auto, elements
</ sql :query >
</ root >

```

附录 2:

```

< script language = 'JavaScript ">
var
ActiveXObject ( 'Microsoft. XMLDOM ' ) :
xmlDoc. async = 'false ";
var
ActiveXObject ( 'Microsoft. XMLDOM ' ) ;
xslDoc. async = 'false ";
xslDoc. load ( 'zwdzqk. xsl ' ) ;
var x ;
var bb = 0 ;
function getCookie ( name )
{
}
function book ()
{
x = 'library/ mylib/ zwdzqk. xml ?';
var y = getCookie ( 'leibie ' ) ;
var z = getCookie ( 'heirong ' ) ;
x = x + y + " = " + z ;
xmlDoc. load ( x ) ;
Layer1. innerHTML = xmlDoc. transformNode ( xslDoc ) ;
}

```

附录 3:

```

function searchtext ( t , x )
{
var root , j ;
var xmlSearchChildNodes ;
var searchxml = newActiveXObject ( 'Microsoft.
XMLDOM ' ) ;
searchxml. async = 'false ";

```

```

bb = 1
if ( x ! = "" )
{
searchxml. load ( 'blank. xml ' )
root = searchxml. selectSingleNode ( ' / root ' ) ;
if ( t = = 'title ' ) { j = ' / root / zwdzqk / Title " ; }
if ( t = = 'sponsor ' ) { j = ' / root / zwdzqk / Sponsor " ; }
if ( t = = 'publisher ' ) { j = ' / root / zwdzqk / Publisher " ; }
if ( t = = 'subjectwords ' ) { j = ' / root / zwdzqk / Subject
words " ; }
if ( t = = 'keywords ' ) { j = ' / root / zwdzqk / Keywords " ; }
if ( t = = 'ISSN ' ) { j = ' / root / zwdzqk / ISSN " ; }
if ( t = = 'classnumber ' ) { j = ' / root / zwdzqk / Classn
umber " ; }
if ( t = = 'description ' ) { j = ' / root / zwdzqk / Description " ; }
xmlSearchNodes = xmlDoc. documentElement.
selectNodes ( ' / root / zwdzqk ' ) ;
xmlSearchChildNodes = xmlDoc. documentElement.
selectNodes ( j ) ;
for ( i = 0 ; i < xmlSearchNodes. length ; i + + )
{
if ( xmlSearchChildNodes. item ( i ) . text. indexOf ( x ) > = 0 )
{
tempNode = xmlSearchNodes. item ( i ) . cloneNode ( true ) ;
root. appendChild ( tempNode ) ;
}
}
Layer1. innerHTML = searchxml. transformNode
(xmlDoc) ;
}
else { alert ( '请输入筛选条件 ! ' ) ;
}

```

参考文献

- 1 孟小峰等. WEB 查询技术研究. 计算机科学, 2001, 28(2)
- 2 肖延松等. 基于 XML 的数据交换在分布式 WEB 系统中的应用. 电脑与信息技术, 2002, 10(1)
- 3 卢啸龙. XML 完全实例教程. 北京: 电子工业出版社, 2001
- 4 飞思科技研发中心. SQL SERVER 2000 XML 应用开发. 北京: 电子工业出版社, 2000
- 5 <http://www.2tigers.net>

谈春梅 南京邮电学院图书馆副研究馆员. 通讯地址: 南京市. 邮编 210003.

段卫华 南京邮电学院图书馆硕士. 通讯地址同上.

刘伟 南京邮电学院应届毕业生. 通讯地址同上.

(来稿时间: 2002-07-30)