

SRW 服务的设计与实现

田明君 杨晓江

(南京师范大学教育技术系, 南京 210097)

摘要 SRW 是一种基于 Web Service 的网络联机检索协议, 它为我们通过 Web 提供信息检索服务, 提供了类似于 Z39.50 的检索能力, 从而提供了基于 Web 的书目资源整合方法。实现一个 SRW 服务虽然比实现一个 Z39.50 服务要相对简单, 但其中会涉及一些新的概念、标准和技术问题, 例如, SOAP 通信架构, CQL 检索语法, Dublin Core 元数据标准等。本文描述了一个 SRW 服务系统的设计思想及具体实现。对涉及的关键技术, 如 SRW 的三个主要操作的执行、SOAP 消息的内容及封装、CQL 语法的解析、MARC 数据与 Dublin Core 元数据的转换等进行了详细的介绍。

关键词 SRW 网络信息检索 Web Service Z39.50

Design and Implementation of SRW Service

Tian Mingjun and Yang Xiaojang

(Department of Education Technology, Nanjing Normal University, Nanjing 210097)

Abstract SRW is a standard form for Internet union search. It provides for us the similar search ability as Z39.50 by web information searching, to offer the resource method based on the web catalogue. Although it's relative simple to the implementation of SRW service, there are some new notion, standard and technology. For example, SOAP communication structure, CQL query language, Dublin Core metadata etc. This paper describe the Design and Implementation of a SRW service system, It also refers to the solution of some key technology, such as the execution of three main operation, the encapsulation of SOAP XML streams, the parsing of CQL and the transform between MARC and Dublin Core.

Keywords SRW, Internet Information Search, web Service, Z39.50

1 引言

Z39.50 协议是网络信息检索标准, 它支持各种高级检索功能, 解决了异构数据库的互操作性问题, 用户可以使用同一个客户检索程序检索分布于 Internet 上的所有 Z39.50 服务器。但作为网络信息检索标准, Z39.50 的应用并不成功, 主要是因为 Z39.50 协议本身的复杂性, 给实现该服务带来了难度; 而且检索 Z39.50 服务器需要专门的客户端软件。Z39.50 Web 网关使用户可以通过 Web 对

Z39.50 服务进行检索, 但它并没有从根本上解决 Z39.50 的复杂性。Web Service 技术可以建立分布式的跨库检索系统, 它是完全建立在 HTTP 协议之上, 并通过 SOAP 的无状态的通讯架构, 解决了各系统间的互操作问题。但是, 要实现真正的跨库检索, 需要对检索语法、记录格式及通信方式等进行统一的规范。

SRW(Search/Retrieve Web Service)^[1] 是基于 Web Service 的网络信息服务标准, 它运用 Web Service 技术将 Z39.50 的精髓应用到因特网通信协议上。SRW 克服了 Z39.50 协议的复杂性, 提炼出 Z39.50

收稿日期: 2007 年 2 月 14 日

作者简介: 田明君, 女, 1980 年生, 硕士生, 南京师范大学教育技术系, 研究方向: 数字图书馆与信息检索。E-mail: tianmingjunyt@163.com。杨晓江, 男, 1965 年生, 研究生导师, 教授, 主要研究方向: 数字图书馆, 情报与文献学, 计算机应用等。

标准中重要的操作,例如,解释操作(Explain)、浏览操作(Scan)、查询操作(SearchRetrieve)等。它以 Web Service 作为技术框架,详细规定了基于 Web 进行信息检索的一组标准操作及其相关参数、检索语法、返回记录格式及信息传递方式等,不但为不同检索系统的集成提供了统一的调用接口,而且使用户可以方便的通过 Web 进行图书馆的书目检索。

SRW 协议为图书馆信息资源的共享提出了理想的解决方法。目前,国外很多大型图书馆都已经支持 SRW 服务,例如,美国国会图书馆在 VoyagerZ39.50 在线目录服务器上借助 IndexData 公司的 YAZProxy 服务器实现了一个 SRW/Z39.50 网关,提供 SRW 标准的解释与查询服务功能;OCLC Research 开发了 SRW 服务器对外提供书目检索服务^[2]。国内对于 SRW 的研究与开发也刚刚开始,对开发该服务需要解决的新问题,如 SRW 三个必备操作的执行过程、SOAP 数据封装、CQL 检索语法的解析以及到 SQL 检索语句的转化、MARC 数据与元数据标准 Dublin Core 的转换等,还没有提出系统的解决方案。本文给出了一个具体的 SRW 服务的设计思路及实现方案,对于实现 SRW 服务需要解决的新问题,本文都给出了可供参考的解决方法。

2 相关概念和约定

2.1 SRW 的操作

SRW 提供的主要操作有 3 个。

(1)解释操作:解释操作是用户了解服务器和数据库功能的操作。当发送解释请求时,服务端返回服务器和数据库功能的相关信息。

(2)扫描操作:扫描操作可以帮助用户明确自己的检索主题,从而调整检索词以达到最好的检索效果。当发送扫描请求时,服务端将返回检索结果的主题列表。

(3)查询操作:查询操作是 SRW 最主要的功能,它可以使用户检索远程数据库中的数据。当发送查询请求后,服务端将返回完整的书目数据。

本文的 SRW 服务系统将实现以上所有的操作。

2.2 检索语法 CQL

SRW 使用 CQL^[3](Common Query Language)作为检索语法。CQL 表达式符合 BNF^[4](Backus-Naur Form)范式的标准,由一个包括布尔运算符、左操作数、右操作数的三元组组成。操作数也可以是一个

三元组或检索子句。检索子句可以包括索引项、关系和检索词。

CQL 中定义了四个布尔操作符 and、or、not、prox。其中,prox 是查询相邻近的记录,可由布尔修饰符(Boolean Modifiers)进一步限定。CQL 中的关系运算符 all、any、=、within、enclose,本文都对其进行了扩展。例如,当关系“all”的检索词中包含多个词的时候,可以将它们扩展成布尔运算符“and”的表达式。关系“within”的检索词为一个范围,检索属性对应的值需要在此范围之内。

本文通过一个 CQL 解析器将 CQL 检索语句映射到 SQL 检索系统。为了防止用户输入不符合 BNF 范式的查询语句,本文在检索界面上对检索语句的格式进行了规范,减轻了服务器的处理负担。

2.3 记录语法格式

SRW 检索记录使用 XML 为记录的语法格式,常用的符合 XML 语法格式的元数据有 Dublin Core、MarcXml、XPath 等。Dublin Core 元数据不仅内容简洁,含义明确,而且它的 15 个元素与 MARC 数据字段的对照也有明确的定义。本文中书目数据库的记录以 MARC 流数据形式存放,因此,本文使用 Dublin Core 元数据标准。MARC 流数据需要通过 MARC 转换器转换成 Dublin Core 格式,然后返回给用户。

2.4 信息传递方式 SOAP

SRW 通过简单对象访问协议 SOAP 传递消息,客户端可以直接发送 SOAP 格式的数据包,由 Web 应用程序将其发送到服务端;也可以 Get 方式发送 URL 或 Post 方式发送表单,由 Web 应用程序将其封装成 SOAP 消息,再发送到服务端。本文采用第二种方案。

2.5 配置文件

配置文件是 SRW 服务系统中非常重要的组成部分,它架起了服务系统与 Web 应用程序之间互相沟通的桥梁。主要包括 SRW 服务相关参数的配置和书目数据库的配置。SRW 服务参数配置文件中定义 SRW 服务系统的具体参数,主要是对服务版本、SRW 各功能支持情况、及各书目数据库配置文件路径及文件名的定义。书目数据库配置文件中主要包括数据库系统支持的语法,例如,XmlSchemas = dc;检索属性与本地检索属性映射关系及数据库其他属性等。

2.6 书目数据库

为了便于说明问题,本文采用文献[5]中的书目数据库设计方案。

3 SRW 服务的总体架构及处理流程

3.1 系统总体结构

系统整体采用 B/S 架构,分为用户层、Web 应用层、Web 服务层和数据层。系统结构示意图如图 1 所示。

上图中,用户通过浏览器发送请求,Web 应用层(即 SRW Servlet)接收请求并将请求封装成 SOAP 消息,发送给远程的 Web 服务(即 SRW Service),Web 服务解析 SOAP 消息,执行相应的操作,将结果封装成 SOAP 消息,返回给 Web 应用层,通过浏览器显示。

3.2 总体处理流程

系统的核心在于 SRW Service 模块所提供的服务,包括:解释,扫描及查询服务。这些服务的总体处理流程如下:

(1)用户通过浏览器发送 Post 请求,请求中包括相关的参数,主要包括:要进行操作的数据库名称,操作的类型及执行操作的相关参数;

(2)SRW Servlet 接收请求,并得到用户请求的相关参数;

(3)SRW Servlet 从 SRW 配置文件中读取各数据库配置文件的文件名和存放路径;

(4)将(2)(3)中得到的信息封装成 SOAP 数据包发送到远程的 SRW Service,即 Web 服务模块;

(5)SRW Service 执行用户请求的操作,然后将

结果封装成 SOAP 数据包返回给用户;

(6)客户端得到 SOAP 数据包,将数据包加上 XML 文件头,并采用 XSLT 将 XML 转换为 HTML 格式,由浏览器显示。

3.3 操作的执行

解释操作和查询操作的具体处理步骤如下:

3.3.1 解释操作

当用户发送的请求中没有检索词时,将该操作视作解释操作。当服务端接收到解释请求时,将从指定的数据库配置文件中获得该数据库的相关信息。需返回的信息包括:①服务器支持协议的版本信息 ②返回记录的编码方式 ③服务器的地址 ④数据库的简要说明 ⑤返回记录的语法格式等。这些信息都可以从 SRW 服务配置文件和数据库的配置文件中获得。

3.3.2 查询操作

SRW 最主要的功能就是检索远程数据库中的数据。

(1)发送查询请求:当进行检索服务时,用户端发送一个查询/检索请求,请求中包含数个检索参数。如:查询语句(query)、一次返回的记录数(maximumRecords)、起始记录(startRecord)、结果集保存时间(resultSetTTL)、记录的语法格式(recordPacking)等。其中,查询语句 query 是查询请求中最重要的参数,它包含了一个代表查询的 CQL 字符串。Web 应用层接收请求并将其包装成 SOAP 消息发送到 Web 服务层。

(2)处理查询请求:

1) Web 服务层从请求信息上下文中获得各数

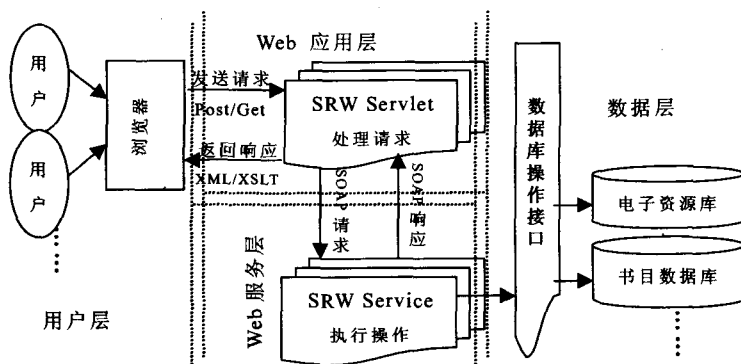


图1 SRW 系统总体结构图

数据库操作类的类名、数据库配置文件的文件名和存放路径;

2) 从各数据库配置文件中读取数据库属性信息(包括检索词本地属性与 Dublin Core 检索点的映射关系等);

3) 从请求的 SOAP 消息中获得查询操作的各参数;

4) 将 query 参数其提交给 CQL 解析器进行分析,如果 query 以“dc.”开头,则查询数据库,产生新的结果集,返回结果集名称,当请求的数据库为多个时,需要对结果进行合并,然后从结果集中提取记录;如果 query 以“cql.resultSetId”开头,则从数据库中提取指定结果集中的数据;当提取数据时,如果结果集已被删除,则返回错误诊断信息 diagnostics;

5) 将记录中的 MARC 流数据提交给 MARC 转换器,将其转换成 Dublin Core 格式;

6) 将结果以 SOAP 消息返回,返回 SOAP 消息中的主要参数有返回结果集的记录总数(numberOfRecords)、结果集名称(resultSetId)、返回查询结果(records)、下一条记录位置(nextRecordPosition)等。

(3)发送提取结果请求:当用户需要从已有的结果集中提取结果(例如,已经执行查询请求,单击“下一页”时,返回已有结果集中的记录),该请求与查询请求最主要的区别在于 query 参数,此时该参数将包括对应的结果集名称。例如,〈srw: query〉cql.resultSetId = * * * 〈/srw: query〉,其中“* * *”为第一次查询时返回的结果集的名称。

3.3.3 扫描操作

扫描操作与查询操作有很多相似之处,它们的区别在于:扫描操作只返回记录的主题信息,而查询操作返回记录全部信息。

4 关键技术设计

4.1 SOAP 消息的内容及其封装

SOAP^[6](简单对象访问协议)消息是 SRW 的信息传递方式,用户向 Web 应用层发送的查询参数将通过 SOAP 封装器封装成 SOAP 消息。SOAP 消息由 Envelope(信封)和 Body(消息主体)组成。

一个查询请求 SOAP 消息主体的例子如下:

```
<soap:Body><srw:searchRetrieveRequest>
  <srw: query>dc.title = 计算机 or dc.title = 电脑</
```


4.2 CQL 语法的解析

CQL 查询语句的分析及对应的 SQL 语句的构造是影响查询准确性和效率的关键因素。由 BNF 范式可知,CQL 可能是包含递归式的复合查询,因此,本文采用两种处理策略。如果查询语句只包含一个查询项,则直接将其转化为对应的 SQL 语句。否则,就将复合查询语句构建成为一颗查询树,该查询树的结构是:①查询树的非叶节点由左子树,布尔运算符和右子树组成;②左子树可以是叶子节点也可以是一个复合查询结构,右子树为叶子节点;③叶子节点是包含索引集、关系和检索词的三元组。根据查询树的结构,本文采用中序遍历查询树,递归地取出查询树中的每一个单查询节点,将单查询节点转换成 SQL 语句然后由布尔操作符连接生成 SQL 查询语句。

在递归取出每一个单查询节点的同时,将该节点转化为相应的 SQL 语句。每一个单查询节点包括索引集及检索属性、关系和检索词,其中,索引集的检索属性需要映射为各书目数据库的本地属性。本文在服务端为各书目数据库建立了 DC 检索属性

与各数据库本地属性映射表,一个检索属性可能需要对应多个本地属性,一个本地属性也可以对应多个检索属性。当索引集的属性不能直接与本地属性对应时,将把它映射成为 MARC 的字段/子字段,对于不同的 MARC 数据(如:CNMARC、USMARC 等)将做不同的映射。映射表的一部分如下所示:

Index.dc.term = 1

Index.dc.creator = 4,5

Index.dc.format = CN :307 \$ a;US :586 \$ q

Index.dc.title = 1,2,3

上表中,等号左侧表示检索集及其属性,右侧表示数据库的本地属性代码或 MARC 字段/子字段。由于 SRW 可以支持多种索引集,如:DC、CQL、BATH、NET 等,因此,我们将映射表存放在配置文件中,可以随时增加对各种索引集的支持,方便的进行扩展。当服务系统启动时,映射表立即被读入内存,当接收到查询请求时,可以快速的将其中的查询属性转换成各数据库本地属性。

4.3 MARC 数据的转换

MARC 数据是目前书目数据库中书目信息的主要存放形式,而 CNMARC 和 USMARC 是最常见的 MARC 格式。服务器必须将查询到的记录数据从 MARC 数据转换成 XML 格式,然后封装到 SOAP 消息中进行传递。本文采用 Dublin Core 元数据表示书目信息,在服务端设计了一个 Marc/Dublin Core 转换器。首先,在服务端的 SRW 配置文件中建立 Dublin Core 与 MARC 字段的映射关系^[7],将 Dublin Core 的 15 个主要元素映射到不同类型 MARC 的相应字段^[8]。MARC 与 Dublin Core 映射表的部分如下所示。

MARCToDC, dc: creator = CN:200 \$ f,700 \$ a;
US:245 \$ c...

MARCToDC, dc: title = CN:200 \$ a;US:245 \$ a
...

MARC 数据的同一个信息可能存放在多个字段中,因此,需要将 Dublin Core 各元素映射到所有与该信息有关的 MARC 字段/子字段。当在某一个字段/子字段中没有取到信息时,可以继续到其他的字段/子字段获取信息。此映射表也将提前读入内存,以便使用的时候能够快速获得。

Marc/Dublin Core 转换器包括 MARC 解析器和转换器两部分,其中转换器是主要部分。MARC 解析器的功能是取出 MARC 数据字段/子字段的内容,不

同类型的 MARC 数据均可以使用同一个 MARC 解析器进行解析。为了将 MARC 数据转换成 Dublin Core 的各元素,MARC 转换器将循环执行以下步骤:①从映射表中读取 Dublin Core 元素对应的 MARC 字段/子字段;②调用 MARC 解析器取出 MARC 字段中的相应信息;③根据映射关系将取出的信息封装成 XML 格式。如果不需要将 15 个元素的信息全部返回,可以根据需要增减映射表中的映射关系。转换以后的部分书目数据如下所示:

<dc:creator>张三</dc:creator>

<dc:title>计算机</dc:title>...

在解析器查找相应字段信息的过程中,需要不断地扫描 MARC 数据字段/子字段,如果查找的字段比较多则会影响返回记录的速度。为了尽量减少扫描次数,本文采取了以下解决方案:先将映射关系中的字段进行排序,然后依次扫描 MARC 中的字段,当在映射的第一个字段中取到信息,则停止扫描,否则,继续从上次扫描结束的字段开始,扫描映射的第二个字段,依次类推。这样,每取得 Dublin Core 的一个元素,只需对 MARC 数据扫描一次,大大提高了系统的检索效率。

5 系统实现

本系统采用 JAVA 语言开发,Apache + Tomcat + SQL Server 为系统支撑平台,采用 Apache SOAP 项目的最新版本 Axis 作为 Web Service 的运行环境。

系统的访问及结果界面简洁友好,访问界面上列出了查询所需的各参数,系统的访问界面和返回结果页面如图 2、图 3 所示。

图 2 系统访问界面

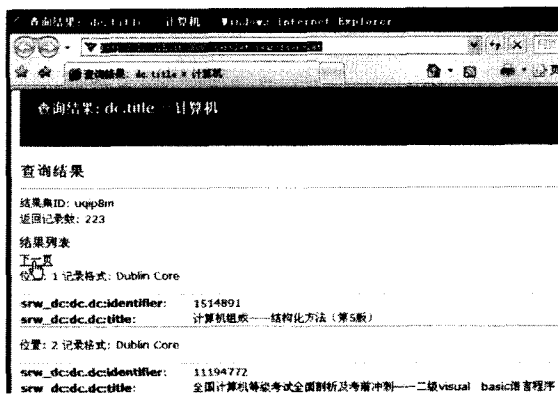


图3 返回结果界面

访问界面列出的主要参数有检索项,一次返回记录数、记录集保存时间、返回记录开始位置以及要查询的数据库等。用户可以方便的对这些参数进行设置。除检索词以外,各参数都设置了默认值,最大限度的减少用户输入参数的个数。用户还可以根据需要动态地添加查询项。由于大多数用户不熟悉 CQL 语法,我们严格按照 CQL 语法格式列出检索项中的各组成部分(索引集、索引属性、关系、检索词和布尔关系),让用户只需选择下拉列表中的项目;而不必关心具体的格式。

6 结束语

SRW 标准促进了深层次的数字资源和服务的

共享,为建立广泛的,开放的,分布式的统一检索提供了标准和策略。该系统的实现对图书馆信息检索的网络化发展有着重要意义。目前 SRW 协议还在进一步完善中,它的对数据库书目信息进行更新的 Update 操作也刚出台不久。另外,SRW 如何与现有图书馆系统及其他图书馆协议进行整合也是下一步需要研究方向。

参 考 文 献

- [1] SRW: Search/Retrieve Web Service. [2007-01-15]. <http://www.loc.gov/standards/sru/srw/index.html>.
- [2] 李聪,胡伟. SRW 的发展和现状分析[J]. 晋图学刊, 2006, 2(93): 30-32.
- [3] Common Query Language. [2007-01-15]. <http://www.loc.gov/standards/sru/cql/>.
- [4] Backus-Naur_form. [2007-01-15]. http://en.wikipedia.org/wiki/Backus-Naur_form.
- [5] 杨晓江,张福炎. 基于 Z39.50 的联机书目检索系统[J]. 软件学报, 1999, 10(8): 824-827.
- [6] Simple Object Access Protocol (SOAP) 1.1. [2007-01-15]. <http://www.w3.org/>.
- [7] MARC to Dublin Core Crosswalk. [2007-01-15]. <http://www.loc.gov/marc/marc2dc.html>.
- [8] CALIS 联机合作编目中心. [2007-01-15]. <http://162.105.139.101/CALIS/lhml/lhml.asp>.

(责任编辑 王建平)