应用实践

DSpace 系统嵌入式检索服务设计与实现*

吴登禄[1,2] 祝忠明[2] 马建霞[2] 韩 柯[3]
[1](中国科学院研究生院 北京 100049)
[2](中国科学院国家科学图书馆兰州分馆 兰州 730000)
[3](华北水利水电学院信息工程系 郑州 450011)

【摘要】介绍选择 SRU 实现 DSpace 系统检索服务的原因和背景,分析基于 SRU 协议的 DSpace 系统嵌入式检索服务的设计架构,并以 JavaScript HTML 和 Google Gadget 的方式实现检索服务,使得 DSpace 检索服务能够很好地嵌入用户环境,与其他系统实现无缝集成。
【关键词】DSpace 嵌入式检索服务 SRU Google Gadget 集成
【分类号】G250

Design and Implementation of Embedded Retrieval Service for DSpace Repository System

Wu Denglu[1,2] Zhu Zhongming[2] Ma Jianxia[2] Han Ke[3]
[1](Graduate University of Chinese Academy of Sciences, Beijing 100049, China)
[2](The Lanzhou Branch of National Science Library, Chinese Academy of Sciences, Lanzhou 730000, China)
[3](Department of Information Engineering, North China Institute of Water Conservancy and Hydroelectric Power, Zhengzhou 450011, China)

【Abstract】This paper introduces the background of SRU(Search and Retrieve via URL) protocol and the reasons why the SRU is chosen as retrieval service for DSpace repository system. More specificially, the authors describe the architecture of the embedded retrieval service for DSpace repository system and build the retrieval service with JavaScript HTML and Google Gadget. It can integrate and embed with other system conveniently.
【Keywords】Embedded retrieval service for DSpace repository system SRU Google Gadget Integration

1 DSpace 嵌入式检索服务需求分析

在 Web2.0 环境下,信息环境变得越来越开放,信息资源将是泛在的,用户需要的信息服务也将是泛在的。在这种背景下,数字图书馆的信息服务模式将超越以图书馆为中心的服务组织模式,向以用户为中心的服务组织模式转变。因此,数字图书馆应用系统的设计和开发,必须考虑如何更加方便地被集成到、嵌入到用户的信息空间和工作流程中去,强调以用户为中心,鼓励用户参与和贡献,使用户以更加习惯和方便的方式参与到系统中。

收稿日期:2008-07-10
收修改稿日期:2008-09-01
*本文系中国科学院国家科学图书馆"全院联合机构仓储体系建设"项目及国家社会科学基金项目"机构知识库建设研究与应用"(项目编号:07BTQ019)的研究成果之一。

DSpace[1]作为一种数字仓储系统，其框架是一种传统型的功能性架构。因此，从用户使用系统的角度来看，用户想要获得其资源和服务必须进入 DSpace 系统环境；从系统集成和数据互交换的角度来看，它只提供部分数据互交换的功能，如基于 OAI－PMH 协议的数据收割服务，在当前分布异构的环境中，它与其他系统实现无缝集成和数据交换的支持方面也存在相当大的局限。那么，如何使 DSpace 系统的功能和服务方便地嵌入到用户的信息空间和工作流程中以及实现数据的互交换呢？

事实上，Andy Powell 早在 2005 年 11 月就提出了基于服务的仓储系统，认为仓储系统可以由存储服务、删除服务、检索服务、收割和获取服务等 5 种服务组成[2]。但对这些服务的实现方面，目前主要集中在存储服务方面。如 JISC（Joint Information Systems Committee）提出的轻量级存储协议 SWORD[3]（Simple Web－service Offering Repository Deposit）和客户端工具方式的实现，可以使用户在 DSpace 系统之外完成数据的提交；有关 DSpace LNI[4]（Lightweight Network Interface）的研究，通过利用 WebDAV 协议来实现对 DSpace 的资源进行访问和实施多种操作；Fedora 系统则提供了基于 Web Services 的两种类型的服务接口，即基于 SOAP 的 Web 服务接口和 RESTful（Representational State Transfer）的 Web 服务接口。对 DSpace 检索服务的研究，主要是 OCLC 的 SRU[5]提供了针对 DSpace 检索服务的实现和支持。

但从总体来看，上述有关仓储的服务接口的实现，主要以面向系统或应用之间的集成为主。OCLC 的 SRU 对 DSpace 检索服务的实现，也是为了解决系统之间数据互交换和面向编程人员的，普通用户很难使用。本文提出了基于 Web Services 的 DSpace 嵌入式检索服务的解决方案，与上述实现不同的是，在利用 Web Services 实现了 DSpace 检索功能之上，以符合 Web2.0 的方式对检索功能和服务进行二次封装和发布，使得非技术用户可以通过简单的脚本或代码粘贴的方式将其嵌入到自己熟悉的工作环境和过程中去。具体的设计思路为：对 DSpace 系统目前功能性的架构进行解构，并主要针对 DSpace 系统的检索功能利用 Web Services 技术进行再次封装和发布，以支持其检索服务被方便集成和嵌入。之所以采用基于 Web Services 的技术解决方案，一方面，由于 Web Services 技术已经成为解决分布、异构环境下系统集成和数据互交换的一种主流技术；另一方面 Web Services 采用 XML 在系统之间交换数据，提供异构平台之间的标准化、结构化的数据传输和交换方法。因此，选择 Web Services 技术可以很好地对 DSpace 系统的检索功能以 Web 服务的形式进行封装和发布，以实现将其向用户的工作流程和信息空间的嵌入。

本文结合研发工作实践，实现了基于 SRU 的 DSpace 系统嵌入式检索服务，并以 JSHTML（Javascript HTML）和 Google Gadget 方式对检索服务进行发布，以支持面向用户环境的嵌入式应用。

2 DSpace 嵌入式检索服务架构设计

DSpace 嵌入式检索服务实现的核心包括两个方面：一是以 Web Services 的方式对 DSpace 的检索功能进行封装，并对外提供统一接口；二是对 DSpace 检索服务接口运用 Web2.0 的技术进行封装，以支持面向用户的嵌入式检索服务。

在对 DSpace 检索功能以 Web Services 方式进行封装的过程中，主要考虑采用基于 SRU 检索服务的方式来实现。SRU 是对 Z39.50[6]检索协议的简化和基于 Web Services 方式的扩展，目前已经成为 Web 信息检索服务的标准。它提供了解决分布、异构数据资源检索、集成和共享的符合 Web Services 方式的解决方案，实现了 Web 查询的标准化、查询结果的结构化，因而具有良好的开放性、易用性，在信息检索服务领域得到了广泛的应用。

SRU 按照消息封装格式标准的不同分为两类：基于 SOAP 格式的 SRU 和基于 URL 格式的 SRU。前者又称为 SRW，在 SOAP Envelope 形式的 XML 文档中，填入信息检索的请求和应答，以 HTTP POST 方式发送请求。后者以 URL 格式对信息检索请求进行编码，并通过 HTTP 的 GET 方法发送请求。SRU 使用 CQL[7]作为查询式的构造语法，与 SQL 和 XQuery 等查询机制相比，CQL 更加简单，用户可以直接记住 CQL。检索结果以基于特定元数据格式（如 DC）的 XML 文档方式返回。

由于基于 URL 的 SRU 把检索式写进 URL，更容易在 Web 环境中实现与其他系统的无缝集成和嵌入，所以，比 SRW 的方式得到更多的选择和应用。这也是本

文选择 SRU 而不是 SRW 来作为 DSpace 系统嵌入式检索服务设计和实现的重要原因之一。

DSpace 嵌入式检索服务框架包括三部分：DSpace 系统资源层、SRU 接口层和嵌入式应用层，如图 1 所示：

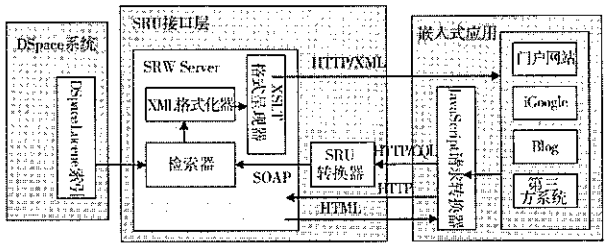

图 1　DSpace 嵌入式检索服务架构

2.1　DSpace 系统资源层

DSpace 系统实现数据的存储，使用 Lucene[8] 索引数据，提供数据的浏览和检索功能，并且为 SRU 层提供公共的检索 API，支持检索服务。

2.2　SRU 接口层

负责对 DSpace 系统资源层的公共检索 API 进行封装，并为应用层提供统一规范的检索接口，包括 SRU 检索转换器、检索器、XML 格式化器、XSLT 格式呈现器。SRU 检索转换器负责将用户提交的包含查询表达式的 URL 请求转换为 SOAP 包格式；检索器收到用户提交的或经过转换器生成的 SOAP 包，执行消息解析，生成查询式，通过 DSpace Lucene 索引，与 DSpace 系统交互，获取检索结果；XML 格式化器负责对接收到的检索结果，按照指定的元数据格式（如 DC）封装成 XML 格式，提供给 XSLT 格式转换器。XSLT 转换器按照默认的转换模板或用户自定义的模板对 XML 结果文件进行转换，并呈现给用户。

2.3　嵌入式应用层

该层提供对 SRU 接口层提供的检索服务按照 Web 2.0 可嵌入式应用发布的要求，进行再次封装和发布，以使任何非技术用户都可以方便地对 DSpace 系统的检索服务进行调用和嵌入。系统提供两种嵌入方式：

（1）以 JSHTML 的方式来发布检索服务，普通用户只需要在自己的 Blog、网站等用户环境中加入一段 JSHTML 代码就可以嵌入 DSpace 系统的检索服务；

（2）利用 Google 的 Gadget 技术，将 DSpace 的检索服务以第三方应用方式发布到 iGoogle 中。用户只要拥有 iGoogle 的账户就可以定制 DSpace 的搜索服务，及时获得机构库中的信息。

DSpace 系统层与 SRU 接口层共同构成胖后台服务器体系，来支撑检索服务，其内部实现过程对用户是不可见的。应用层构成瘦客户端系统，通过 SRU 提供的检索服务，应用层可以很方便地集成 DSpace 系统的搜索服务，不必关心其中的实现细节[9]。

3　系统实现

3.1　SRU 接口封装

DSpace 检索功能的封装，采用了 OCLC 的开源工具 SRW/U[10]，它是一个基于 Java 语言的开源系统。DSpace 系统也是一个基于 Java 语言的系统，相同的开发语言环境使两者之间的集成较容易实现。基于这一开源软件，为 DSpace 系统提供 SRU 封装接口的过程中，主要进行了如下方面的定制和扩展：

（1）全文检索支持的扩展：SRU 对 DSpace 系统资源的检索，是通过检索 DSpace 的 Lucene 索引来实现的。默认条件下，它仅支持包括按作者、标题、关键字等字段的检索，不支持全文检索。本文在实现过程中，通过对其进行扩展，以支持全文检索。

（2）返回结果集的定制：该软件包，在某些方面显然留有比较明显的测试实现的痕迹（如在结果的返回方面，就只能够返回结果集的部分记录，不能全部返回检索结果集中的记录）通过对其修改和定制，可以根据需要返回结果集的全部记录或一定数量范围的结果集记录。

（3）中文支持的优化：尽管 SRU 软件包在特殊字符（如：“-”、“/”、“=”等）以及 Unicode（ISO 10646）编码字符支持方面有相对较好的支持，但在中文字符的支持方面还存在乱码问题。通过对属性名、值进行 UTF-8[11] 编码，允许汉字等编码出现，以支持中文检索。

（4）检索服务方法的定制：SRU 的核心是一个 Web 服务引擎，SRU 的 Web 服务定义了三种方法：Explain 方法、SearchRetrieve 方法和 Scan 方法，其中，SearchRetrieve 方法是 SRU 服务的核心，它执行对数据库的查询操作并返回结果集。在 DSpace 的数据检索服务实现中，只使用 SearchRetrieve 方法，来实现 DSpace 系统的嵌入式检索服务。

3.2　服务请求接口

SRU 以 URL 的方式来对外提供 DSpace 数据检索

接口，其检索请求的 URL 由三部分组成：主机地址、服务路径、检索参数。以已经扩展了 SRU 接口的 SeeKSpace[12] 系统为例，进行简要的说明：

（1）主机地址，即 DSpace 系统的主机地址，在本例中为：

http://seekspace.resip.ac.cn。

（2）服务路径为：/SRW/search/DSpace?。

（3）检索参数为：

query = dc.subject + = + " water" &version = 1.1&operation = searchRetrieve&recordSchema = info:srw/schema/1/dc - v1.1&maximumRecords = 100&startRecord = 1&resultSetTTL = 300&recordPacking = xml&recordXPath = &sortKeys = 。

在上述查询式中，参数 version 和 query 是必备项，其他是可选项，其中 query 是基于 CQL 的检索表达式，各主要参数的含义如表 1 所示：

表 1　查询参数

名称	类型	必备	说明
query	字符串	是	用 CQL 描述的检索表达式
version	字符串	是	表示客户端支持的版本号，一般为 1.1
operation	字符串	否	操作类型，这里固定为 SearchRetrieve
recordSchema	字符串	否	结果数据采用 XML 模式的 URI
maximumRecords	字符串	否	客户端要求 response 当前页返回的最大记录数。
startRecord	整数	否	客户端要求 response 返回的第一条记录的位置
resultSetTTL	整数	否	结果集缓存的时间
recordPacking	字符串	否	指定命中结果返回的格式，可以是 String 或是 XML
recordXPath	字符串	否	确定记录格式的 XML 格式路径
sortKeys	字符串	否	指定排序方式，可以对多个字段进行排序

其中，检索参数详细说明如下：

① query：用 CQL 描述的检索表达式。例如上面的 dc.subject + = + "water"，系统提供的可供查询的字段包括：标题（title）、关键词（keywords）、作者（author）、描述（description）、全文（all）。

② version：表示客户端支持 SRU 的版本号，服务器端可以按照客户端的要求发送消息，如果服务器端的版本号高于客户端的版本号，那么服务端将降低版本号，适应客户端的要求。

③ operation：描述当前的操作类型，可选的操作类型有 Scan，Explain，searchRetrieve。在 DSpace 检索服务的实现中只使用 SearchRetrieve。

④ recordSchema：客户端要求返回的记录需要遵循的基于 XML 的元数据记录格式，默认为 DC。

⑤ maximumRecords：系统返回给客户端的最大记录数，取值应大于等于 0。

⑥ startRecord：说明客户端要求 response 返回的第 1 条记录的位置。服务器端可能检索到若干条记录，用户如果只想取其中的一部分，则可使用该参数。取值应大于 0。默认值为 1。

⑦ resultSetTTL：结果集在服务器缓存的时间，在此系统设置的默认值是 300 秒。

⑧ recordPacking：指定返回的结果记录的封装格式，可以是 String 或是 XML。

⑨ recordXPath：用来标识 xsl 的 xpath，这个参数只适用于 1.1 版本。

3.3　检索服务应答格式

DSpace 系统服务器按照客户端的查询请求，以 XML 或者 String 的形式返回查询应答，如表 2 所示：

表 2　查询应答参数

名称	类型	必备性	说明
version	字符串	是	服务器 SRW 协议的版本，一般为 1.1
numberOfRecords	整数	是	命中记录的数量。查询失败时 numberOfRecords 的值为 0
resultSetId	字符串	否	用来标识当前查询结果的一个唯一标识符
resultSetIdleTime	整数	否	服务器端认为结果集能够保留的时间
records	记录序列	否	返回的记录结果列表
nextRecordPosition	下一记录	是	返回下一记录的位置
diagnostics	诊断序列	否	错误信息列表
extraResponseData	XML 片段	否	扩展 ResponseData
echoedSearchRetrieveRequest	字符串	否	将查询信息按简单 XML 格式返回给请求方

其中 records 是最核心的参数，包括 recordSchema、recordPacking、recordData、recordPosition、extraRecordData 和 nextRecordPosition。客户端发送检索请求，服务器返回 XML 的数据。其中，recordData 包含了所有命中记录的元数据；recordPosition 记录了命中记录在结果集中的位置；nextRecordPosition 指示了下一次返回记录的位置，适用于翻页的情况。

下面以 seekSpace 系统为例，以“ecology”为检索词，检索表达式如下所示：

http://seekspace.resip.ac.cn/SRW/search/DSpace? query = data.all + % 3D + % 22ecology% 22&version = 1.1&operation = searchRetrieve&recordSchema = info%3Asrw%2Fschema%2F1%2Fdc - v1.1&maximumRecords = 10&startRecord = 1&resultSetTTL = 300&recordPacking = xml&recordXPath = &sortKeys =

返回的 XML 数据如下：

<? xml - stylesheet type = " text/xsl" href = "/SRW/searchRetrieveResponse.xsl"? >

< searchRetrieveResponsexmlns = " http://www.loc.gov/zing/

```
srw/" >
<version > 1.1 </version >
<numberOfRecords > 605 </numberOfRecords >
<resultSetId > agtbbp </resultSetId >
<resultSetIdleTime > 300 </resultSetIdleTime > <records >
  ……
<dc:format. extent > <! [CDATA[43 bytes]] > </dc:format.
    extent >
<dc:format. mimetype > <! [CDATA[text/plain]] > </dc:for-
    mat. mimetype >
<dc:language. iso > <! [CDATA[en]] > </dc:language. iso >
<dc:subject > <! [CDATA[ecology(生态学)]] > </dc:sub-
    ject >
<dc:subject > <! [CDATA[ecosystem ecology(生态系统学)]]
    > </dc:subject >
<dc:subject > <! [CDATA[community ecology(群落生态
    学)]] > </dc:subject >
<dc:subject > <! [CDATA[population ecology(种群生学)]]
    > </dc:subject >
<dc:subject > <! [CDATA[conservation ecology(保护学)]] >
    </dc:subject >
<dc:subject > <! [CDATA[evolutionary ecology(进化学)]] >
    </dc:subject >
<dc:title >
  <! [CDATA[Acta Oecologia - International Journal of Ecolo-
    gy]] >
</dc:title >
<dc:title. alternative >
  <! [CDATA[生态学报:国际生态学杂志]] >
</dc:title. alternative >
<dc:type >
  <! [CDATA[Journal]] >
</dc:type >
  ……
</srw_dc:dc >
</recordData >
<recordPosition > 1 </recordPosition >
</record >
<record >
  ……
```

3.4　嵌入式服务实现

运用 SRU 接口层提供的检索接口，以 JSHTML 和 Google Gadget 的方式来发布 DSpace 嵌入式检索服务。

(1)以 JSHTML 实现服务嵌入

以 JSHTML 实现 DSpace 系统服务的嵌入，即直接将 SRU 检索请求通过 JSHTML 嵌入到用户的环境中(比如博客、个人空间、iGoogle 等)，从而使用户不必关心底层的实现。用户只要在 HTML 页面中嵌入 DSpace 系统提供的一小段脚本代码，就可以很容易地将 DSpace 的资源与服务嵌入到用户工作流程的某一环节上。以 JSHTML 方式实现检索服务嵌入的核心 JavaScript 代码如下：

```
<SCRIPT TYPE = "text/javascript"
  SRC = "http://seekspace. resip. ac. cn /test/test. js" >
</SCRIPT >
<SCRIPT TYPE = "text/javascript" >
  SearchEngine( );
</SCRIPT >
```

上述代码的核心是 SearchEngine( )函数，它负责建立与 DSpace 系统的连接。用户只要在自己的网页中嵌入上述 JavaScript 代码，SearchEngine( )函数将自动请求 DSpace 系统，建立连接，并返回检索界面。返回检索界面的核心代码如下：

```
out. println( " < SCRIPT LANGUAGE = \" JavaScript \" SRC =
  \". /scripts/
external. js\" > </SCRIPT > " );
out. print( " </SCRIPT > " );
out. print( " </HEAD > " );
out. print( " < BODY > " );
out. print( "  < form action = /cgi - bin/post - query method = \"
  post\" > " );
out. print( " Input Search Text" );
out. print( " < input type = \" text\" name = \" test\" value = \" ht-
  tp://\" / > " );
out. print( " < br / > " );
out. print( " < input type = \" button\" name = \" search\" value
  = \" submit\"
onclick = \" RetrieveDirect( )\" / > " );
out. print( " < br / > " );
out. print( " </form > " );
out. print( " </body > " );
out. print( " </html > " );
```

函数 RetrieveDirect( )负责对用户的检索词和 DSpace 的 SRU 检索接口进行对接，并请求 DSpace 系统返回检索结果。

(2)以 Google Gadget 实现服务

一个基本的 Gadget 有如下所示的结构：

```
<? xml version = "1.0" encoding = " UTF - 8" ? >
  < Module >
```

```
<ModulePrefs title = " DSpace Retrieve" / >
<Content type = " html" >
  <! [CDATA[
    //包含上述 Javascript 的 HTML 代码
  ]] >
</Content >
</Module >
```

其中,XML 描述 Gadget 的结构、HTML 和 CSS 提供表示层、Javascript 提供 Gadget 的逻辑层。

< Module > 标签指示这个 XML 文件包含了一个 Gadget; ModulePrefs 属性指明了 Gadget 的标题,如 DSpace Retrieve; < Content type = " html" > 行表示 Gadget 的内容类型是 HTML; CDATA 部分包含 HTML 或 Javascript 代码,这些代码用来提交和激活具有特定功能的 Gadget,如具有天气信息的 Gadget、RSS 阅读器 Gadget 等。

DSpace 检索服务 Gadget 的实现,就是在以上结构的 CDATA 部分加入前述 DSpace 检索服务的 Javascript 代码,就可以构造出能够在 iGoogle 等环境中可发布的 Gadget。具体的发布过程为:上传 DSpace 到 Web 服务器,登录 iGoogle 个性化主页,通过点击右上角的"增加内容(add stuff)"链接加入 Gadget。这样就可以实现 DSpace 嵌入式检索服务。用户通过登录 iGoogle,就可以检索和发现 DSpace 检索服务 Gadget,将其定制和融入到自己的 iGoogle 定制环境中,以无需登录到 DSpace 系统的方式,直接检索和获取 DSpace 系统提供的资源和服务。

4 结 语

通过为 DSpace 加装基于 SRU 的 Web Services 接口,并以 JHTML 和 Google Gadget 的方式将这一检索服务以符合 Web 2.0 的应用方式进行发布,为以 DSpace 系统为基础的数字知识库服务在面向用户的信息环境和工作过程的嵌入和集成方面,提供了简捷的技术解决方案,在促进 DSpace 系统及其资源和服务的应用扩展方面起到积极的作用,同时也极大地方便了用户对相关资源和服务的获取。

参考文献:

[1] DSpace[EB/OL]. [2008 - 07 - 08]. http://www.dspace.org.

[2] A 'Service Oriented' view of the JISC Information Environment [EB/OL]. [2008 - 09 - 01]. http://www.ukoln.ac.uk/distributed - systems/jisc - ie/arch/soa/.

[3] SWORD[EB/OL]. [2008 - 09 - 01]. http://www.ukoln.ac.uk/repositories/digirep/index/SWORD.

[4] DSpace LNI[EB/OL]. [2008 - 09 - 01]. http://wiki.dspace.org/index.php/LightweightNetworkInterface.

[5] SRU:Search and Retrieve via URL(standards, Library of Congress) [EB/OL]. [2008 - 07 - 08]. http://www.loc.gov/standards/sru/index.html.

[6] Z39.50[EB/OL]. [2008 - 07 - 08]. http://www.z3950.org/.

[7] Common Query Language(SRU:Serach and Retrieve via URL - Standards, Library of Congress) [EB/OL]. [2008 - 07 - 08]. http://www.loc.gov/standards/sru/cql/index.html.

[8] Lucene[EB/OL]. [2008 - 07 - 08]. http://lucene.apache.org/java/docs.

[9] 李春旺,王小梅,王昉,等.基于 SRU 的集成服务平台设计与实现[J].*现代图书情报技术*,2007(10):12 - 15.

[10] OCLC[EB/OL]. [2008 - 07 - 08]. http://www.oclc.org/asiapacific/zhcn/research/software/srw/default.htm.

[11] RFC3986. Uniform Resource Identifier(URI):Generic Syntax[EB/OL]. [2008 - 07 - 08]. http://rfc.net/rfc3986.html#s2.1.

[12] seeKSpace[EB/OL]. [2008 - 07 - 08]. http://seekspace.resip.ac.cn.

(作者 E - mail:wudl@llas.ac.cn)