●张付志 肖 阳

数字图书馆包装层生成技术研究*

摘 要 包装层是对应于特定信息源的一种特殊程序,是实现异构信息源集成的关键。数字图书馆的包装层可以作为数字图书馆的智能前端代理,负责把来自中介层的统一查询表示映射为针对具体数字图书馆的查询格式,并从得到的查询结果中提取出用户需要的信息,提交给中介层。数字图书馆包装层生成的关键技术主要有:查询映射机制,查询服务调用机制,结果提取与转换机制。图 6。参考文献 5。

关键词 数字图书馆 包装层 查询映射 程序生成器
分类号 G250.76

ABSTRACT The authors introduce the definition of wrapper level and its roles in digital library, and also analyze key technologies for the generation of wrapper lever of digital library, such as query mapping. 6 figs. 5 refs.

KEY WORDS Digital library. Wrapper level. Query mapping. Program generator.
CLASS NUMBER G250.76

用户为了获得需要的资料,往往需要访问多个数字图书馆,同一查询请求不得不重复提交。更多的情况下,用户希望数字图书馆能够提供统一的查询服务,一次性获得所需的全部数字图书馆资源,而无需适应多种系统与查询界面。中介层/包装层结构无疑是解决这种异构数字图书馆资源整合的有效途径之一。

包装层的生成经历了手工编写、半自动化生成以及目前正在兴起的自动化生成 3 个阶段[1]。大多数研究都是采用基于规则的生成方法,并且在生成过程中绝大部分代码是用来处理查询和数据转换的。因此,对于包装层生成技术的研究主要是对查询映射算法和信息抽取规则的研究。

本文利用 XML 作为中间信息交换格式,通过建立一种查询能力描述模型来讨论查询映射算法[2],并利用程序生成器技术来实现数字图书馆包装层程序的半自动化生成。

1 数字图书馆包装层结构模型

中介层/包装层结构为用户提供了一个可以对多个异构信息源进行访问的统一接口,它为实现 Web 上异构数字图书馆系统的集成提供了一条有效途径。基于中介层/包装层结构的数字图书馆系统集成框架如图 1 所示。包装层负责将中介层提供的统一查询表示转换成数字图书馆使用的查询格式,然后接收每个数字图书馆返回的查询结果,并从中提取出用户需要的信息,最后将结果转换成中介层可以处理的统一数据格式,返回给中介层。中介层通过包装层对不同的数字图书馆进行包装,隐藏了数字图书馆的异构性,为用户提供可以访问多个数字图书馆的统一界面,提高了对数字图书馆访问的透明性。

数字图书馆包装层主要由查询映射机制、查询服务调用机制和结果提取与转换机制 3 部分组成,其结构模型如图 2 所示。查询映射机制负责将来自中介层的查询请求转换成数字图书馆的查询服务程序所使用的查询格式;转换后的查询请求通过 HTTP 协议(HTTP POST/GET 请求)传送到数字图书馆的服务器端,调用数字图书馆的查询服务程序执行查询操作,并接收该查询服务程序所返回的查询结果;最后由结果提取与转换机制负责提取出用户所需要的信息,并以统一的数据格式提交给中介层。数字图书馆的查询映射与数据转换结构如图 3 所示。

* 本文得到河北省教育厅基金项目(200206)、燕山大学博士基金项目和河北省科技研究与发展计划(05213583)资助。

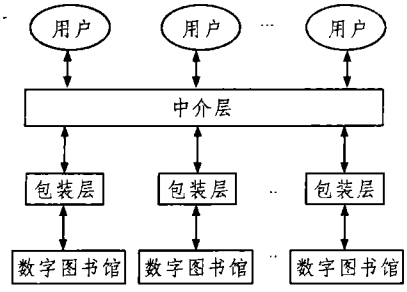

图1　基于中介层/包装层结构的数字图书馆系统集成框架

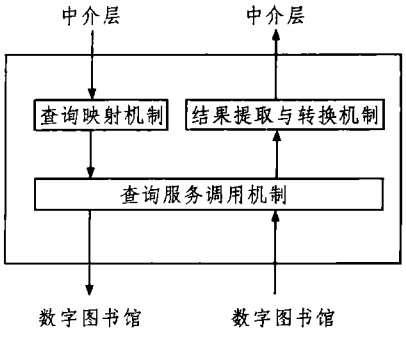

图2　数字图书馆包装层的结构模型

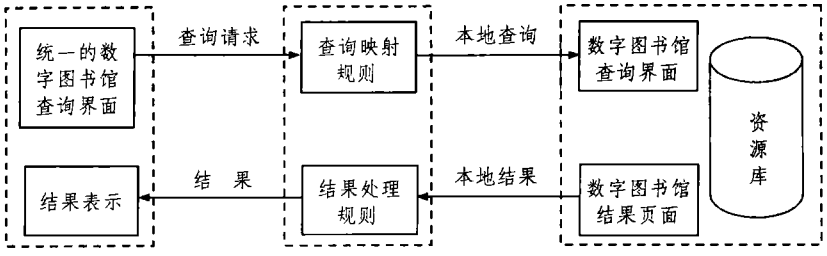

图3　数字图书馆的查询映射与数据转换结构

2　数字图书馆包装层生成的关键技术

数字图书馆包装层的生成主要有两种方式:一种是完全依靠手工编写包装层程序代码来实现,即由设计人员根据对数字图书馆查询界面的分析,手工编写程序代码;另一种是通过自动方式来实现,即利用人工智能中的机器学习技术,通过对样本数字图书馆查询界面的学习,自动地或者在用户参与下半自动地生成包装层程序代码。显然,前一种不仅效率低,而且维护工作量大。另外,随着网上数字图书馆的不断增加和访问界面的更新,这种方式也缺乏灵活性。因此需要采用自动或者半自动的方式来生成数字图书馆的包装层程序。

2.1　查询映射机制

2.1.1　查询能力描述模型

查询映射是数字图书馆包装层生成中的关键与难点[3~4]。为了实现中介层的统一查询表示与包装层所对应的数字图书馆的本地查询之间的映射,需要对各自的查询能力进行描述。为此,本文提出一种具有适应性的查询能力描述模型,并通过该描述模型来获取中介层的统一查询界面和数字图书馆的查询界面相对应的查询能力。

数字图书馆的查询界面控件可分为3种类型:分类控件、查询输入控件和结果显示控件。查询输入控件是最重要的,它在很大程度上影响着查询映射的实现。该控件可细分为:查询输入关键词、关键词修饰词和逻辑操作符。图4、图5给出了IEEE的高级查询界面和某中文检索界面的快照。

查询输入关键词就是用户键入文本输入框中的内容;关键词修饰词(Modifier)包括域修饰词和限定修饰词两部分。域修饰词(Field)用来限制关键词出现的范围,比如Keyword, Abstract等,限定修饰词(Qualifier)则用来描述关键词的形式,比如exact phase, exactly like等;通过逻辑操作符来连接两个查询输入关键词,比如and, or等。

基于以上概念,图4、图5所对应的查询能力描述模型如下:

$Q_{IEEE} = \{T_1\theta_1T_2\theta_2T_3\}$

Field(Ti) ∈ { < All Fields >, < Abstract >, < Document Title >, ……},$1 \leqslant i \leqslant 3$;

$\theta i \in \{$and, or$\}$, $1 \leqslant i \leqslant 2$

$Q = \{T\}$

Field(T) ∈ {搜索文章标题和正文,搜索标题,搜索正文}

Qualifier(T) ∈ {匹配所有关键词、匹配整个短

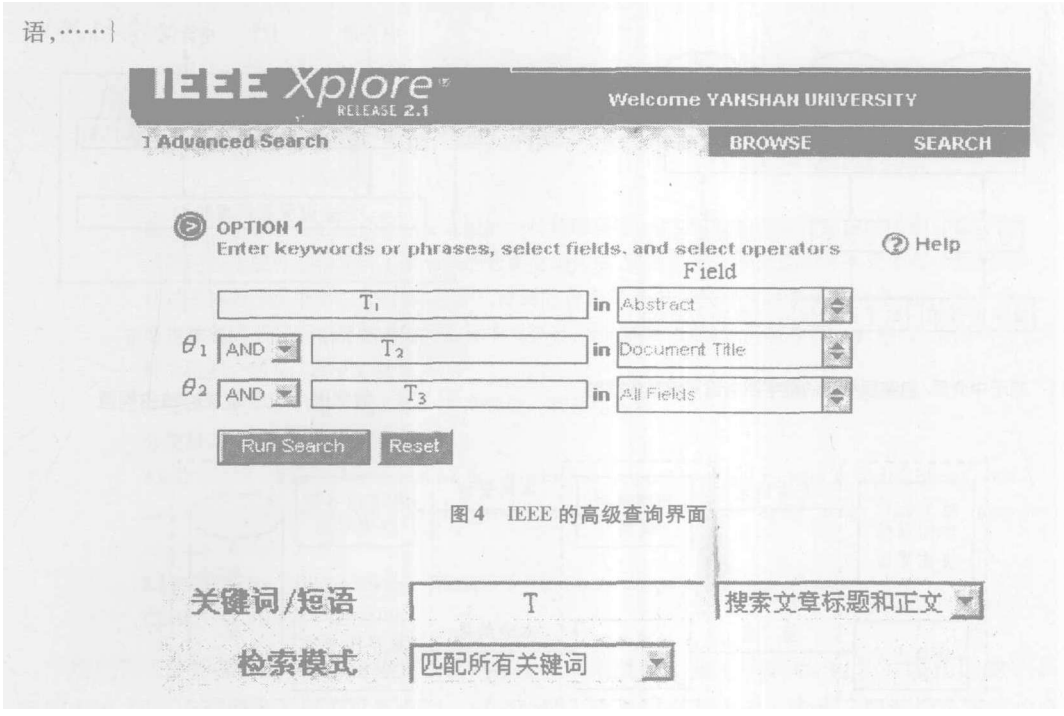

图4 IEEE的高级查询界面

图5 某中文检索界面

2.1.2 查询映射算法

在给出查询映射算法之前,应先明确两个相关的概念。

查询分解:将一个联合的查询表示分解成几个联合的子查询表达式(在子查询表达式中关键词之间通过"and"∧,或者是"not"¬ 逻辑操作符进行连接)或者是几个单个的关键词的操作。

由于中介层所提供的统一查询界面的查询能力与数字图书馆所提供的查询界面的查询能力之间存在着差异,用户的查询请求有可能不能完全满足,需要一个查询过滤机制来尽可能地为用户提供满意的查询结果。

查询过滤:利用那些没有被完全推送到数据源的剩余查询条件来对查询结果做进一步处理的操作。

下面是对该算法的简要描述。

算法名称:查询映射

输入:用户提交到中介层的查询表示(M)和目标数字图书馆的查询能力描述模型(W)

输出:目标数字图书馆所支持的转换后的查询表示(Q)和查询过滤条件(F)

①Vector $M_{mediator}$ = decompose(M);//将M进行查询分解后所产生的联合子查询表达式存储在$M_{mediator}$中

②for(int i = 1; i < = N_m, i + +)//N_m为M分解后所产生的联合子查询表达式的数目

③m = $M_{mediator}$[i];//获取M分解后的一个联合子查询表达式

④Boolean Sendflag = FALSE;//用来确定一个子查询表达式是否可以被推送给目标数字图书馆

⑤for(int j = 1; j < = NumberOf(m); j + +)//函数NumberOf()用于获取查询能力描述模型中关键词的数量

⑥for(int k = 1; k < = NumberOf(W); k + +)

⑦if(Modifier(T_j^m) ∈ Modifier(T_k^W))

⑧$T_j^m \rightarrow T_k^W$;//关键词的映射

⑨Modifer(T_j^m)→Modifer(T_k^W);//关键词修饰词的映射

⑩$\theta_j^m \rightarrow \theta_k^W$;//逻辑操作符的映射

⑪Q←new(m,W);//函数new(m,W)生成新的子查询表达式并将结果存放在Q中

⑫end if;

⑬end for k;

⑭SendFlag = TRUE;

⑮end for j;

⑯if(SendFlag = FALSE)

⑰redecompose(M);

⑱end for i;

⑲if(NotEmpty($M_{mediator}$))

⑳F←$M_{mediator}$;//获取用于查询过滤的剩余查询条件

㉑return Q,F;

在上述查询映射算法中,首先对来自中介层的查询表示 M 进行查询分解,并将结果存放在变量 $M_{mediator}$ 中,然后从中选取一个联合的子查询表达式与"底层"数字图书馆所对应的查询能力描述模型进行查询映射。随着映射操作的进行,将新产生的、目标数字图书馆所支持的子查询表达式存储在变量 Q 中,当映射操作完成后,将变量 $M_{mediator}$ 中剩余的子查询表达式存放在变量 F 中,用于查询过滤。

其中,中介层与"底层"数字图书馆之间的映射操作部分见算法中的⑤~⑬步,即对关键词及其所对应的修饰词以及逻辑操作符进行映射操作;如果 m 与 W 之间是不匹配的,就对 M 做进一步分解(如算法中的⑰所示),并重新进行映射操作,直到产生新的、为目标数字图书馆所支持的子查询表达式为止。

2.2 查询服务调用机制

数字图书馆的包装层与数字图书馆之间通过 HTTP 协议进行通信,通过发送 HTTP POST/GET 请求来调用数字图书馆的查询服务程序。

对于某个确定的数字图书馆来说,包装层在 HTTP 请求中提供了执行查询服务程序所需要的参数,如:查询关键词、关键词的修饰词等。数字图书馆的查询服务程序则根据其包装层所提供的参数执行查询操作,并将查询结果作为 HTTP 响应信息,返回给发出请求的数字图书馆包装层。

2.3 结果提取与转换机制

包装层的另一个重要功能是完成结果的提取与转换。每个数字图书馆返回的查询结果页面中都或多或少地包含一些头部和尾部信息,这些信息通常对用户没有太大的价值。为了能够向中介层提供统一的查询结果格式,要求数字图书馆的包装层对接收到的查询结果页面进行分析,从中提取出对用户有用的信息,并将结果转换成一种中介层可以处理的统一数据格式。

实际上,结果提取的过程就是对返回的 HTML 网页字符串进行处理的过程。因此,可以利用对字符串操作的方法来截取出对用户有用的信息。

3 基于 XML 和 Java 的数字图书馆包装层半自动生成

3.1 数字图书馆的 XML 描述

XML 是一种完全可移植的数据格式,用户可以根据需要定义 XML 标签。本文采用 XML 来描述数字图书馆所对应的信息、即包装层所对应的数据源的信息。

数字图书馆的 XML 描述包括 3 种信息:数字图书馆的普通信息、查询映射信息以及结果提取与转换信息。

数字图书馆的普通信息包括该数字图书馆的搜索地址和搜索方法;查询映射信息包括数字图书馆的访问方法,以及该查询界面与统一的查询界面间的映射信息等;结果提取与转换信息包括对返回的结果页面进行提取的信息以及最后提交给中介层的数据表示的信息等。下面是 IEEE 高级查询界面的 XML 描述框架:

```
<DL wrapper>
    <search_DL title = "search info:" >
    <search_method > POST </search_method >
    < search _ URL > http://ieeexplore. ieee. org/
search/advsearch. jsp </search_URL >
</search_DL >
< query_mapping >
< query string > …… </query string >
……
</query_mapping >
……
</DL Wrapper >
```

3.2 数字图书馆包装层生成器

程序生成器就是能够生成程序的程序[5]。设计一个程序生成器意味着不仅仅是编写一个程序,而是要写一个可以生成许多程序的程序。由于采用手工方式编写数字图书馆包装层程序不仅效率低,而且缺乏灵活性,为此提出一种基于 XML 和 Java 的数字图书馆包装层程序半自动生成方法,并利用 Java 语言实现一个简单的数字图书馆包装层生成器。其结构模型如图 6 所示。该程序生成器主要由 XML 语法分析器、DOM 的分析与转换和 Java 代码生成器 3 部分组成。它能够根据所建立的 XML 描述文档,生成相应的数字图书馆包装层 Java 程序源代码。

数字图书馆包装层生成器需要使用一个标准的 XML 语法分析器。本文采用 Apache Xerces 来实现,以便对读入的 XML 文档进行语法分析,并转换成 DOM 数据结构。分析与转换可以直接在创建的 DOM 数据结构上进行。DOM 数据结构存储在内存中,Java 代码生成器可以直接从 DOM 数据结构中获取信息,实现数字图书馆包装层 Java 程序源代码的生成。考虑到程序代码的复用,可将(下转第 64 页)

模块在查询语句的表达上还需要进一步融入XQuery、XPath等XML检索语言。我们也将在此基础上研究基于XML的相关反馈、异构XML数据的处理转化与评价问题等。

致谢：

感谢国家留学基金委资助本文第一作者访问伦敦城市大学从事XML检索的相关研究工作，也感谢Stephen Robertson教授和Andrew Macfarlane博士的悉心帮助和指导。

参考文献

1,6 N. Govert and G. Kazai. Overview of the Initiative for the Evaluation of XML retrieval(INEX)2002. Proceedings of the 1st Workshop of the Initiative for the Evaluation of XML Retrieval(INEX). 2002,1 - 17

2 Introduction of Okapi. Available via http://www. dotty. soi. city. ac. uk/ ~ andym/OKAPI-PACK/.

3 Robertson, S. E. and Sparck Jones, K. Relevance weighting of search terms. Journal of the American Society for Information Science, 1976, 27: 129 - 146.

4,5 Robertson, S. E. Overview of The OKAPI Projects. Journal of Documentation, 1997, 53(1)

7,9,10 Robertson, S. E. and Steve Walker. Some simple effective approximations to the 2-Poisson model for probabilistic weighted retrieval. Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval. 1994, 345 - 354.

8 Robertson, S. E., Hugo Zaragoza and Michael Taylor. Simple BM25 Extension to Multiple Weighted Fields. CIKM 2004. ACM Press. 2004, 42 - 49.

陆 伟 博士，副教授。通信地址：武汉大学信息资源研究中心。邮编430072。

夏立新 博士，华中师范大学信息管理系教授，华中科技大学管理学院博士后。通信地址：武汉华中师范大学信息管理系。邮编430079。 （来稿时间：2005 - 12 - 09）

（上接第55页）用于生成包装层的公共代码，如包装层与网络的连接操作等代码放入公共代码库中。

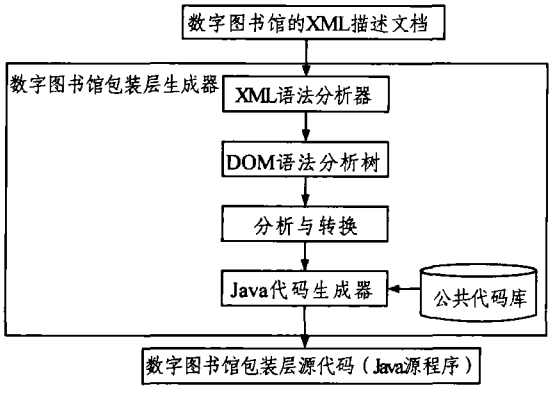

图6 数字图书馆包装层生成器的结构模型

参考文献

1 Yue-Shan Chang, Min-Huang Ho, Wen-Chen Sun, et al. Supporting unified interface to wrapper generator in integrated information retrieval. Computer Standard & Interfaces, 2002, 24(4)

2 Lieming Huang, Matthias Hemmje, Erich J. Neuhold. ADMIRE: an adaptive data model for meta search engines. Computer Networks, 2000, 33: 431 - 448

3 Chen-Chuan K. Chang, Hector Garca-Molina. Mind Your Vocabulary: Query Mapping Across Heterogeneous Information Sources. in : Proc. SIGMOD 99, Philadelphia, PA, June 1999. pp. 335 - 346

4 Donald Kossmann. The State of the Art in Distributed Query Processing. ACM Computing Surveys, 2000, 32(4)

5 冯少荣. 基于XML和JAVA构建程序生成器. 计算机应用与软件, 2005, 22(1)

张付志 燕山大学信息科学与工程学院博士，教授。通信地址：河北省秦皇岛市。邮编066004。

肖 阳 燕山大学信息科学与工程学院硕士研究生。通信地址同上。 （来稿时间：2005 - 11 - 08）