

文章编号:1671-9352(2007)11-0077-05

一种基于 Web Services 的数据库 资源服务化方法

张峰

(山东科技大学 信息科学与工程学院, 山东 青岛 266510)

摘要:基于 Web Services 技术,提出了一种将数据库资源服务化的方法,通过 Web 服务提供数据库的访问接口。给出了服务注册的服务元数据模型,实现了服务的自动发现,并介绍了根据用户请求自动生成 SQL 语句的方法,在数据集成系统中提供了透明的数据访问。

关键词:Web Services;数据库;服务封装

中图分类号:TP311 **文献标志码:**A

An approach to building services of database based on Web Services

ZHANG Feng

(School of Information Science and Engineering, Shandong University of
Science and Technology, Qingdao 266510, Shandong, China)

Abstract: Based on Web Services, an approach to building services of database resource was proposed. The Web Services offers interfaces for the database. A service meta data model was introduced which is used when the services are registered, thereby the automatic discovery of services was realized. Then the method of building SQL sentences automatically according to the user's request was put forward. Thus users can access the data transparently in the data integration system.

Key words: Web Services; data base; services encapsulation

0 引言

随着互联网技术的发展,网络上可访问的资源正在迅速的增加,对大量分布式、异构数据资源的集成成为急需解决的问题。数据集成技术旨在为用户提供一个一致的查询界面和透明的数据访问,实现对分布式异构数据的集成^[1]。当前该领域已有一些相关的方法^[2]。近年来,随着面向服务架构(SOA)的提出和 Web Services 技术的发展,基于 Web Services 技术构建数据集成系统已经成为一个重要的研究方向。Web Services 采用开放标准,具有协议的通用性以及与平台、语言无关的独立性,适用于解决分布式异构环境中各个数据源之间的数据交换,因此适用

于异地异构数据源的集成。基于 Web Services 技术构建信息集成系统要解决各类分布式、异构资源的服务化问题,而作为网络环境中的一类最主要的数据资源,数据库资源的服务化成为首要解决的问题。

相关研究基于 Web 服务的输入、输出进行服务描述,根据输入、输出参数的匹配发现服务^[3,4]。但在这些方法中,一个 Web 服务仅提供一类特定数据的查询。例如,根据一个学生信息管理数据库形成的 Web 服务可以根据学号查询学生选课成绩,但对于用户其他的查询要求,则需要生成一个新的服务来满足要求。因此,在实际中为了满足用户的需要,要提供大量的、不同类别的服务,有时则需要多个服务通过组合来满足用户要求,从而增加了实现的难度。

收稿日期:2007-05-26

基金项目:国家自然科学基金资助项目(60603090);国家科技基础条件平台建设资助项目(2005DKA64201);山东省“泰山学者”建设工程

作者简介:张峰(1981-),男,助教,研究方向:工作流技术,面向服务的计算,数据集成.Email: angeltom@163.com

由英国 e-Science 中心开发的 OGSA-DAI 项目试图通过 Grid Services 实现对不同的数据库系统的统一访问^[5,6]。该项目致力于形成一个通过网络访问和集成来自不同数据源的中间件,提供数据访问和集成的统一服务接口,通过这些接口,将分布式、异构的数据源映射为单一的数据资源。在实际的使用中,通过网格数据服务(GDS)实现对某个数据资源的访问,用户通过服务组注册器(DAISGR)找到所需的GDS,然后通过执行文档(perform document)定义在GDS上需要执行的操作。OGSA-DAI 可以为用户提供虚拟表,用户通过虚拟表实现对低层数据的访问。但是,在访问数据库资源时,用户需要通过执行文档表达自己的需求,在形成执行文档时仍需用户了解虚拟表或底层表的物理结构,这样对系统的用户提出了更高的要求,没有为用户提供透明的数据访问。

本文提出了一种基于 Web Services 的数据库服务化方法,给出了服务注册时的元数据模型,不仅使用 Web 服务实现了对数据库的封装,而且可以根据用户请求自动的发现服务、生成服务需要执行的 SQL 语句。用户在使用时只需在交互界面上提供检索要求,不必了解底层信息,实现了透明的数据访问。使用该方法构建的 Web 服务可以满足该服务所封装的数据库中各种类型数据的检索,减少了所需形成的服务的数量,降低了系统的实现难度。

1 背景介绍

数据库服务化的目的是使用 Web 服务屏蔽掉数据库的底层物理信息,通过 Web 服务提供统一的访问接口,从而将该数据库资源接入到一个数据集成系统。在基于全局模式的数据集成系统中,首先由领域专家形成该数据领域内的标准元数据集,基于该元数据集表达用户的数据请求^[7]。

Web 服务形成后,基于服务元数据模型,将服务注册到服务注册中心。在数据集成系统中,根据用户请求和服务的描述信息,自动地在服务注册中心发现可以满足用户检索请求的 Web 服务,由这些 Web 服务将基于标准元数据表达的用户请求转换为需要执行的 SQL 语句,并由 Web 服务执行 SQL 语句,最终返回结果。图 1 给出了基于 Web Services 的数据集成系统的工作原理。

基于 Web Services 技术服务化数据库时涉及到 2 项关键技术:

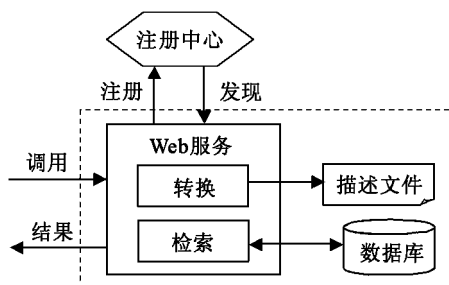


图 1 基于 Web Services 的数据集成系统
Fig.1 Data integration system based on Web Services

(1) 服务元数据模型:将底层数据库服务化后形成的 Web 服务要注册到服务注册中心,但传统的基于 WSDL 的服务描述并不能满足服务发现的要求,因此需要设计一个服务元数据模型,基于该模型完成 Web 服务注册时服务描述信息的手工添加,根据该描述信息实现服务的自动发现。

(2) SQL 语句的生成:响应用户请求过程中,在服务注册中心得到所需调用的 Web 服务后,数据集成系统把用户的请求发送给发现的 Web 服务;Web 服务的转换部件根据描述文件中的信息,将基于元数据集描述的用户请求自动转换为该数据库底层操作的 SQL 语句,最后由 Web 服务的检索部件执行该 SQL 语句完成数据的操作。

下面具体介绍这两项关键技术。

2 关键技术

2.1 服务元数据模型

在文献[3,4]的实现方法中,每个服务的定义本质上都对应该类 SQL 语句,因此一个 Web 服务只能满足一种类型的查询,而为了满足用户不同的需要则要为一个数据库形成多个服务。基于上述认识,本文形成 Web 服务的思路是:将用户的请求作为服务的输入,根据该输入自动生成可以在该服务封装的数据库所在 DBMS 的上执行 SQL 语句,服务的输出是执行 SQL 语句后的结果集生成的 XML 格式的消息。这样,一个数据库上仅需提供一个 Web 服务,减少了生成服务的数量,降低了实现的难度。

然而,基于 WSDL 的服务描述方法难以实现数据库服务化后所形成的 Web 服务的自动发现。因此,需要在 Web 服务注册时手工添加一定的服务描述信息,根据该信息可以实现服务的自动发现。下面给出了服务元数据模型 SMDM(service metadata model),基于该模型实现服务注册时服务描述信息的添加。

服务元数据模型 SMDM 是一个三元组:(SBI,

SSI, SDI)。

(1) SBI(service basic information)表示服务的基本信息,是一个四元组:(SIdentifier, SName, SPublisher, SAddress),其中, SIdentifier 表示服务的全局唯一标识符, SName 表示服务的名称, SPublisher 表示服务的发布者或发布单位, SAddress 表示服务的访问地址。

(2) SSI(service state information)表示服务的状态信息,以标识当前的服务所封装的数据库资源是否可用。

(3) SDI(service data information)表示服务的数据信息,是基于元数据集描述的字段集合。

基于服务元数据模型对 Web 服务进行注册时, SSI 初始化为“可用”,表示可以通过当前的服务访问数据库,在以后的调用过程中可以根据数据库访问状况对该属性进行修改;SDI 是该数据库可以提供的所有基于元数据集的字段集合,通过 SDI 中描述的字段集合与用户请求条件中字段集合的比较可以判断该服务是否符合要求。SDI 描述了服务提供数据的能力,本质上是对服务功能的描述。因此,SDI 是服务自动发现中的关键描述信息。下面给出了服

务发现的规则。

记用户请求中包含的基于元数据集的字段集合为 M ,一个服务的 SMDM 模型中的 SDI 对应的字段的集合记为 N ,那么一个服务可以满足该用户请求的条件是: $M \subseteq N$ And SSI = “可用”。

2.2 SQL 语句的生成

一个 Web 服务被调用时,会从输入参数中得到基于元数据集的用户请求,Web 服务根据用户的请求自动转换成可执行的 SQL 语句。在实现中,为每一个数据库形成一个 XML 格式的描述文件,根据描述文件将用户请求中基于元数据集的字段名称转换为底层数据库各个对象中的字段名称。需要说明的是,一个数据库中包含了表、视图、存储过程等多种对象,通过操作这些对象都可以返回结果。本文当前所形成的 Web 服务的操作仅针对表,因此,描述文件中仅提供了各个表中的字段与元数据集中字段名称的映射关系。

描述文件是以 XML Schema 格式定义的文件,图 2 给出了描述文件在 XML 编辑工具 XMLSpy 中的结构。

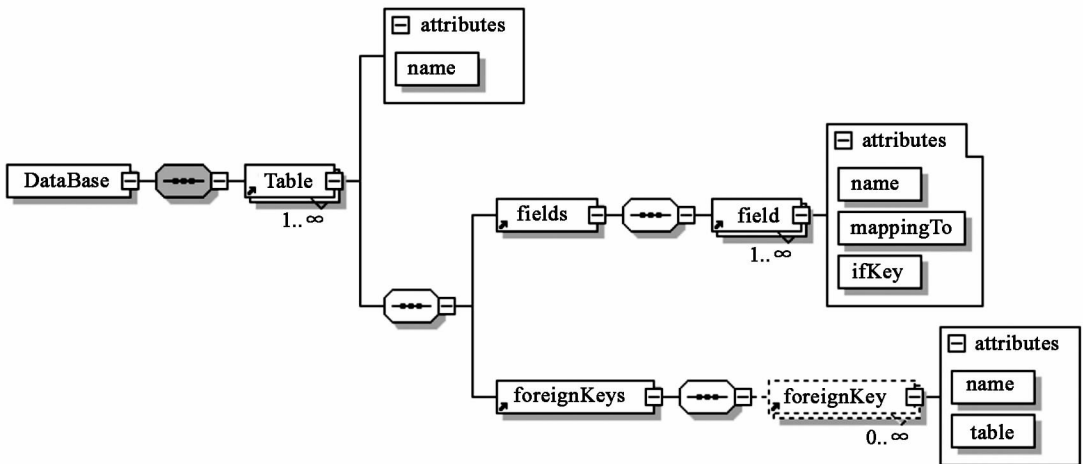


图 2 描述文件结构
Fig.2 Structure of description file

图 2 的结构中对一个数据库的表进行了描述。其中,每个表的属性 name 用于表示表的物理名称;每个表都有若干的 field 子元素,用于表示表的字段,每个字段的 name, mappingTo, ifKey 属性分别表示该字段的物理名称、映射到标准元数据集的字段名称以及“是否为主键”;foreignKeys 记录了每个表的外键元素,foreignKey 有属性 name 和 table,分别对应了该外键字段的名称和以该外键为主键的表的名称。需要说明的是,引入 mappingTo 属性是为了解决数据集成系统中数据源的异构问题,但当前仅限

于对最简单的语法映射的情况,还没有解决复杂的语义冲突问题。

3 应用实例

本章以一个学校的教务管理系统为例介绍数据库的服务化方法。在该教务管理系统的数据库中存在多个表,取其中的 3 个表:学生表、课程表、成绩表作为描述的对象,这 3 个表的关系模型和物理表的结构分别为:

(1) 学生(学号,姓名,性别,出生日期,班级),表的物理结构为:Student(sno, sname, ssex, sbirthday, class),主键为 sno。

(2) 课程(课程号,课程名),表的物理结构为:Course(cno, cname),主键为 cno。

(3) 成绩(学号,课程号,成绩),表的物理结构为:Score(sno, cno, degree),主键为(sno, cno)。

数据集成系统中的标准元数据集对上面3个表包含的属性的对应描述为(StudentNo, StudentName, StudentSex, StudentBirthday, ClassNo, CourseNo, CourseName, Degree)。

步骤1 根据标准元数据集形成对应的描述文件,描述文件的一个片段为:

```

<DataBase>
  <Table name = "Student">
    <fields>
      <field name = "sno" mappingTo = "StudentNo" ifKey = "1"/>
      <field name = "sname" mappingTo = "StudentName" ifKey = "0"/>
      <field name = "ssex" mappingTo = "StudentSex" ifKey = "0"/>
      <field name = "sbirthday" mappingTo = "StudentBirthday" ifKey = "0"/>
      <field name = "class" mappingTo = "ClassNo" ifKey = "0"/>
    </fields>
  </Table>
  <Table name = "Course">
    <fields>
      <field name = "cno" mappingTo = "CourseNo" ifKey = "1"/>
      <field name = "cname" mappingTo = "CourseName" ifKey = "0"/>
    </fields>
  </Table>

```

```

    <foreignKeys/>
  </Table>
  <Table name = "Score">
    <fields>
      <field name = "sno" mappingTo = "StudentNo" ifKey = "1"/>
      <field name = "cno" mappingTo = "ClassNo" ifKey = "1"/>
      <field name = "degree" mappingTo = "Degree" ifKey = "0"/>
    </fields>
  </Table>
  <foreignKeys/>
  <foreignKey name = "sno" table = "Student"/>
  <foreignKey name = "cno" table = "Course"/>
</foreignKeys/>
</Table>
</DataBase>

```

步骤2 编写代码形成服务并发布到 Web 服务容器 Axis。由于每个 Web 服务都是从输入参数中接收用户请求,并将请求根据描述文件转换成需要执行的 SQL 语句,最终将结果返回,因此,所有的服务都有相似的实现编码,降低了服务开发中的复杂度;

步骤3 将发布的 Web 服务注册到服务注册中心。基于服务元数据模型 SMDM,该服务的描述信息如下:

SBI:(s₁, JWDataService, 教务处, http://localhost:8080/axis/JWDataService.jws)

SSI:可用

SDI:(StudentNo, StudentName, StudentSex, StudentBirthday, ClassNo, CourseNo, CourseName, Degree)。

至此,教务系统数据库的服务化完成。下面看一个具体的数据检索实例。

用户在交互界面中输入检索信息的请求,如图3所示:

图3 用户交互界面
Fig.3 User interface

用户提交请求后,系统基于标准元数据集中的字段定义,将检索字段“学号”转化为 StudentNo,将检索字段“课程名”转化为 CourseName,由此所生成

的请求中包含的检索条件为:StudentNo = ‘103’ and CourseName = ‘操作系统’。该请求中包含的基于元数据集的字段集合 M = {StudentNo, CourseName}。根

据服务发现规则,服务 JWDataService 的 SDI 对应的集合中包含了 StudentNo 和 CourseName,并且 SSI 为“可用”,因此服务 JWDataService 可以满足用户的要求,系统调用服务 JWDataService。服务 JWDataService 根据描述文件判断 CourseName 属性可以从 Course 表中得到,而 StudentNo 属性可以从 Student 表和 Score 表中得到,再根据 Score 表和 Course 表的外键关系,转换生成的 SQL 语句为:

```
select *
from Score, Course
where Score.cno = Course.cno and Score.sno =
'103' and Course.cname = '操作系统'
```

最后,服务执行该 SQL 语句返回结果。

根据以上实例可以看出,使用该方法对数据库服务化后,用户对数据检索时只需提供所需检索的数据的信息,服务的匹配、发现、调用、SQL 语句的执行全部对用户透明,因此用户不必了解底层的信息,方便了用户对数据资源的使用。

4 总结和下一步工作

本文给出的一种数据库资源的服务化方法,基于该方法实现了服务的自动发现和调用,为用户提供了透明的数据访问。

在当前的解决方案中,通过 Web 服务仅提供了对数据库中表的访问,对其他的数据对象,如视图、存储过程的访问未提供支持。调用服务时服务之间是独立工作的,还没有提供对多个服务之间协

同工作的支持。SQL 语句的自动生成是一个较为复杂的过程,要在此研究的基础之上给出其公式化的实现方法。这些都是下一步需要重点解决的问题。

参考文献:

- [1] MAURIZIO L. Data integration: a theoretical perspective[C]// Proceedings of the ACM Symposium on Principles of Database Systems(PODS). New York: ACM Press, 2002: 233-246.
- [2] 陈跃国,王京春. 数据集成综述[J]. 计算机科学, 2004, 31(5):48-51.
- [3] 喻建军,杨冬菊,韩燕波. 一种面向城市应急联动问题的资源服务化方法及其支撑工具[J]. 计算机应用研究, 2006,23(2):53-56.
- [4] HANSEN M, MADNICK S, SIEGEL M. Data integration using Web services[C]// BRESSAN S. et al. Proceedings of the VLDB 2002 Workshop Efficiency and Effectiveness of XML Tools and Techniques and Data Integration Over the Web (EEXTT). Hong Kong: Springer2Verlag, 2003, 2590/2003:165-182.
- [5] ANTONIOLETTI M, ATKINSON M P, BAXTER R, et al. The design and implementation of grid database services in OGSA-DAI[J]. Concurrency and Computation: Practice and Experience, 2005, 17(2-4):357-376.
- [6] KARASAVVAS K, ANTONIOLETTI M, ATKINSON M P, et al. Introduction to OGSA-DAI Services[C]// Lecture Notes in Computer Science, Berlin: Springer, 2005, 3458:1-12.
- [7] HALEVY A Y. Theory of answering queries using views[J]. SIGMOD Record, 2000, 29(4):40-47.

(编辑:孙培芹)