

基于 Web Service 柔性异构数据集成系统设计及查询处理

唐智勇¹, 吴 刚²

(1 国防科技大学 C4ISR 技术国防科技重点实验室, 湖南 长沙 410073; 2 上海交通大学 软件学院, 上海 200240)

摘 要: 提出的异构数据集成系统框架包括源模式层、局部模式层、全局模式层和用户模式层. 利用 Web service 接口完成从源模式层到局部模式层的转换, 利用模式融合和冲突解决实现从局部模式到全局模式的转换, 并嵌入数据查询过程. 用户模式层作为全局模式层的一个子集, 可以动态调整, 方便集成数据的查询处理, 还定义了一个基于 XML 的统一数据模型以支持异构数据的集成中的查询处理, 提出了一个相应的查询处理过程.

关键词: 异构数据源; XML; Web Service; 查询处理

中图分类号: TP31

文献标识码: A

文章编号: 1000-7180(2010)08-0036-05

Query Disposal and Design of Dynamically Adjustable Heterogeneous Data Integration Based on Web Service

TANG Zhi-yong¹, WU Gang²

(1 Key Lab of Science and Technology for National Defense of C4ISR Technology, National University of Defense Technology, Changsha 410073, China; 2 School of Software, Shanghai Jiaotong University, Shanghai 200240, China)

Abstract: The heterogeneous data integration architecture of this paper includes four levels: source schema level, local schema level, global schema level, user schema level. We use Web service interface to complete the transformation from the source schema level to the local schema level. We implement the transformation from the local schema level to the global schema level using the mergence of different local schemas and the solution of confliction. The user schema level which can adjust according to the real-time situation dynamically is a subset of global schema level. We also define a XML based uniform data model to support the querying in the integration of heterogeneous data, and propose a query disposal process for the system.

Key words: heterogeneous data source; XML; Web Service; query disposal

1 引言

数据集成技术^[1]的研究始于 1970 年代中期^[2]. 从一开始的多数据库集成发展到现在的异构数据源集成, 数据集成的范围和作用都在不断扩大. 然而异构数据集成中的查询却没有得到很好的解决. 目前的数据查询技术有数据的透明访问技术、分布式对象计算技术、XML 技术和 Agent 技术.

本文针对现在的数据查询问题中, 已有的数据集成系统未能很好地解决底层异构数据源的“即插即用”问题, 全局模式同各数据源模式的转换工作繁杂, 用户查询视图不能很好的自动调节等缺点, 提出

了一种新型的数据集成系统和查询处理方法, 并在该系统中应用了柔性异构数据的查询处理.

2 数据模型

数据模型是解决信息集成系统中不同模式之间异构性的基础, 因此设计一个集成系统的首要问题就是选择一个合适的公共数据模型. 一种设计较好的公共数据模型应足够简单, 易于系统开发者理解、分析和操作^[3-5].

现在的大部分 DBMS 都用面向对象模型作为它们的通用数据模型. 但是随着各种不同应用程序的出现, 需要扩展传统的面向对象数据模型来集成.

各种不同的数据源. XML 已逐渐变成表现不同数据的领域标准,特别是基于网络的信息系统.作为一种自描述的语言,XML 可以描述各种数据结构,比如线性表、树和图.因此,XML 变成一个各种应用程序集成的数据接口的通用规范.

下面给出了在定义数据模型之前必须要首先定义的两个概念——模式和视图.

定义 1 模式 (Schema) 是关于数据的数据,是描述数据的元信息.其内容是数据信息的描述,包括数据的定位信息(记录了数据的来源和目标)、数据语义描述、数据结构描述等,同时还记录了转换的规则.

定义 2 视图 (View) 是满足上面定义的模式的数据.某个视图一定是和相应的模式联系起来的.是最终呈现给用户的数据.

用户所关心的是视图,而模式是给计算机看的,验证视图是否满足要求.我们用 XML 表示视图,用 XML Schema 表示模式.

我们定义三层数据模型为 XBDM (XML Based Data Model),包括根、实体和属性.

根 (Root),全局模式只有一个根,表示在统一的全局模式上进行数据操作的出发点.一个根可以对应于多个实体.对应于数据库中的概念,相当于数据库 (database).

实体 (Entity),一个根可以有多个实体,同一个实体可以由异构数据源上的多个局部模式组成.一个实体如果对应于不同的异构数据源上的不同的局部模式,那么这些局部模式应该是语义等价的,即它们表示同一个东西,对于语义等价的判断过程,需要用户进行交互,是由用户作决定,哪两个局部模式是语义等价的.一个实体可以对应于多个属性,但是只能有一个根.相当于数据库中表 (table) 的概念.

属性 (Attribute),属性可以分为键属性和一般属性,而键属性又包括主键属性和外键属性.其中主键属性只能有一个,可以通过主键属性来唯一地标识一个实体.外键属性是对于其他的主键属性的引用.一般属性就是正常的字段,可以为空或者是有对应的数据.属性是包括数据和数据类型的.一个属性只能是一个实体的子集.属性相当于数据库中字段 (column) 的概念.

三层数据模型中最核心的是实体,我们定义实体数据模型如下:

定义 3 实体节点 (Entity Node) 的定义为: $EN = \langle Name, Key, Attribute \rangle$, Name 是名字, Key

是一组键属性列表,代表主键属性或者外键属性, Attribute 是一组属性列表.其结构图如图 1 所示.

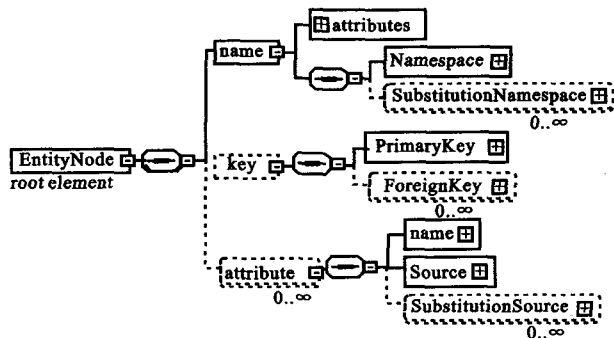


图 1 实体结点的结构图

其中 Name 又包括属性值 (attributes)、命名空间 (namespace)、备用命名空间 (substitution namespace), 分别表示名字的值, 主要名字是从哪里来的, 备用名字是从哪里来的. 命名空间表示主命名空间, 只能有一个; 而备用命名空间则可以有多个. 命名空间和备用命名空间表示的是被集成的模式相关信息, 说明该全局模式是由语义等价的多个局部模式集成得到的, 记录关于被集成的局部模式相关的信息, 比如数据源、表名等. 键又包括主键 (Primary key) 和外键 (Foreign Key). 键其实是一种特殊的属性. 主键、外键以及属性一样, 都包括名字 (name)、源 (source)、备用源 (substitution source). 这里的源和备用源其实都是对名字里的主命名空间、备用命名空间的引用. 可以只有源, 没有备用源; 或者有一条源, 多条备用源. 但是不会出现只有备用源而没有源的情况, 这样的话备用源就要变成源. 源和备用源表示的意思是说, 当数据集成的时候发生冲突的处理办法, 首先以源里的信息为主, 只有当源对应的信息缺失时才会考虑备用源里的数据.

3 系统架构

和大部分异构数据集成系统相似, 我们的系统架构也是基于一种分层的概念^[4-5], 如图 2 所示. 主要的区别是我们的系统是基于 Web service 的, 并且把模式集成和视图集成清楚地区分开来. 在集成过程中, 局部模式被集成引擎所集成, 如图 3 所示, 并生成一个用户和应用程序可用的全局模式, 局部模式的集成是通过集成引擎的工具完成的. 在这个过程中需要用户的交互, 由用户来判断要集成的两个局部模式是否是语义等价的, 从而来决定这两个局部模式是否需要被融合. 最后用户可以在全局模式上的用户模式上查询数据. 模式的集成只做一次, 而

不像其他的系统,每次向系统提交一个查询,就必须在局部数据库系统和数据集成上重构多重查询.在我们的系统中,用户可以基于全局模式来查询,而不用了解分布式异构数据源上的数据信息.全局模式能被增量的重构,只要一个局部模式有了改变,或者一个新的局部模式被加了进来,都可以通过简单的修改从而能够快速的重构全局模式.这种技术和“一次构建,运行多次”的思想一致.这把模式集成从查询中分离了出来,并简化了集成系统的设计,使得系统维护更加有效率.

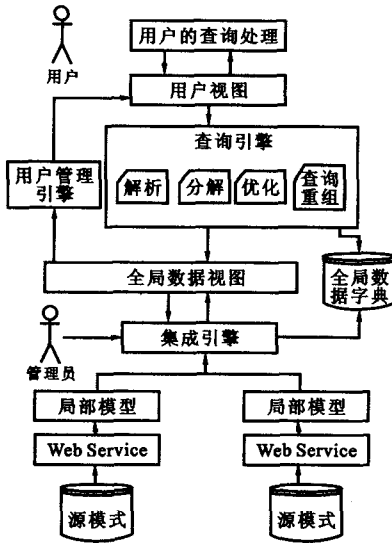


图2 系统架构

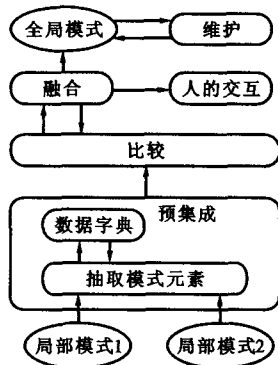


图3 模式集成引擎

系统大体上分为四层:

(1) 源模式层(Source schema level):源模式以局部数据源的局部数据模型表示.因此,不同的局部异构数据源可能以不同的数据模型进行表示,比如:关系模型,面向对象模型,等等.而XML文档是一种基于文档的数据模型,它的源模式是DTD或者XML Schema.

(2) 局部模式层(Local schema level):对于每一

个局部数据源,MDBS应当提供一种工具来自动的把它的局部数据源上相关的部分转化为以我们的通用数据模型表示的局部模式.这种转化会创建从源模式到局部模式的映射.

(3) 全局模式层(Global schema level):基于全局数据模型,通过集成很多局部模式来形成全局模式.它表现了全局视图,以及关于分布的全局数据的映射信息.

(4) 用户模式层(User schema level):因为用户的个性化定制和存取权限等原因,要为不同的用户组和应用程序提供用户视图.

四层架构之间的转换主要有两层,因为源模式层是分布式异构数据源本来就有的,所以从数据源到源模式不需要转化.

源模式到局部模式的转换:基于我们的通用数据模型,通过Web service等方法把源模式转化为局部模式.这个转化只是基于不同的数据模型和模式,是表示方法的转化,并没有冲突的解决,转化前后的名字、值、数据类型等不变.

局部模式到全局模式的转换:把基于局部数据模型的局部模式转换到基于全局数据模型的全局模式.两种模型只是表示的范围不同,表示方法是相同的.这个转换主要包括识别语义一致的本体,融合及冲突的解决.这个过程需要人的参与来识别本体.

全局模式到用户模式的转换:这个转换只是在全局模式上根据用户的定制生成一个子模式,是可以自动完成的.

4 查询转化

查询转换主要分为对查询命令的转换和对查询结果的转换.查询命令的转换指的是输入的基于图形用户界面的查询命令转化到SQL语言查询命令格式和XQuery查询命令格式.查询结果转换是指将异构数据源的数据模型与公共数据模型进行相互转换,从而达到在异构数据源中进行数据查询的目的.

4.1 查询命令转化

在本文的集成系统中,主要考虑了对结构化与半结构化数据的集成.结构化数据一般是指关系数据库的二维表,对它的数据查询主要是SQL语言中的select命令;半结构化数据现在主要是指XML文档,对XML数据集进行查询主要是XQuery^[6]中的FLWOR表达式.所以在设计查询命令转化时,只需考虑把用户输入的基于图形用户界面的查询命令转化到SQL语言查询命令格式和XQuery查询命令格式.

(1) 到 SQL 的映射

如果数据源类型为关系数据库,则在接收到查询引擎传过来的查询命令时,必须将接收到的用 XML 文档格式表示的查询命令转化为检索关系数据库的 SQL 命令形式.在前面设计查询命令文档的词汇集时,元素的选择是参照 SQL 的 select 命令结构形式,即每个词汇都对应着 SQL 查询命令中的某个关键字.所以在进行查询命令格式转换时,只要转换功能模块按照 SQL 查询命令的组成语法顺序读取要查询文档的内容,再加上相应的 select 查询关键字,即可把 XML 文档表示的查询命令转换为 SQL 查询命令形式.

(2) 到 XQuery 的映射

如果数据源类型为 XML 文档,则在接收到查询引擎传过来的查询命令时,必须将接收到的用 XML 文档格式表示的查询命令转化为检索 XML 数据集的 XQuery 命令形式. XQuery 是定义为对 XML 数据集进行查询的,XML 数据集不仅指 XML 文档,还指一切看起来像 XML 的数据,包括关系数据库. XQuery 对于 XML 数据,就像 SQL 对于关系数据一样.而且 FLWOR 表达式在 XQuery 中的功能作用就像 select 语句在 SQL 中的功能作用.

4.2 查询结果转化

信息集成主要是使各异构数据源能相互共享数据,为此在前面我们设计了公共数据模型来实现这个目的,这就需要使各异构数据源的数据模型能与公共数据模型相互转换.在本文的集成系统中,主要考虑两种目前非常流行且重要的数据类型与本文的全局通用数据模型 XBDM 的转换,这两种数据类型为 RDBMS 数据和 XML 数据.对于其他数据类型可以先模拟成这两种数据类型,再与 XBDM 进行转换.这里对查询结果的转化和上面介绍的对查询表达命令的转化是对应的.

(1) 关系模型到 XBDM 模型的转换

在前面章节中我们已经讲述过关系模型与公共数据模型的映射关系,关系模型中的一个二维表对应 XBDM 数据模型中的一个节点数据,而二维表中的每一个属性值就对应节点的键或者属性.公共数据模型 XBDM 是基于 XML 文档标准的,可以很容易地表示成 XML 文档的形式.所以通过公共数据模型 XBDM,关系数据模型可以较容易地转换为基于 XBDM 模型的 XML 文档格式.

(2) XML 文档模型到 XBDM 模型的转换

XQuery 的 FLWOR 表达式查询 XML 文档返

回结果仍为 XML 文档形式,为了使 XML 文档子结果能被组装成全局查询结果,故需将 XML 文档形式转换为公共数据模型 XBDM.因为 XBDM 模型是基于 XML Schema 模型的,在将 XML 文档转换为 XBDM 模型时,只需将基于一种 XML Schema 的 XML 文档转换为基于另外一种 XML Schema 的 XML 文档.所以这种转化可以较容易地完成.

5 查询处理过程分析

5.1 查询处理的特点

对于基于 Mediator/Wrapper 的信息集成系统,查询处理与传统分布式数据库系统的查询处理及普通多数据库系统的查询处理有很大不同.图 4 是集成系统的数据源部分框架简图,通过对此图的分析可以得出此类集成系统的查询处理特点.每个数据源都有三层结构,每个数据源都有统一的中介器层(Mediator),在本文的系统中就是局部模式,而不同数据源也都有不同的包装器层(Wrapper),在本文的系统中就是在每个异构数据源之上定义的 Web Service.

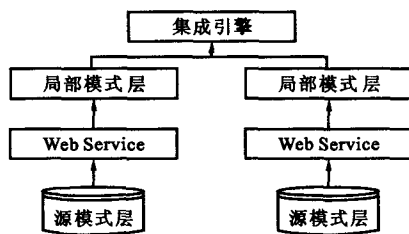


图4 数据源框架图

下面阐述基于 Mediator/Wrapper 信息集成查询处理的特点:(1)用户通过全局视图提出全局查询,查询引擎将全局模式分解为对应各数据源的全局子模式,再由各数据源的 Web Service 将全局子模式转化为自己的局部查询模式,最后集成引擎将各数据源上的 Web Service 返回的子查询结果进行组装返回给用户^[7],所以不需要考虑数据分片及配置等问题。(2)数据源都能相互通信,且每个数据源都能满足彼此的数据处理能力要求.与普通多数据库系统查询相比,为查询优化策略的选择提供了较多灵活性.

5.2 查询处理过程

查询处理过程如图 5 所示.

(1) 查询解析.对于用户的每一个查询请求,首先是进行语法解析.在此过程中,查询中对虚拟视图的引用被元数据管理模块中的视图定义替代,解析器执行静态类型检查,利用元数据管理模块检查查询请求的合法性,判断查询数据是否存在.

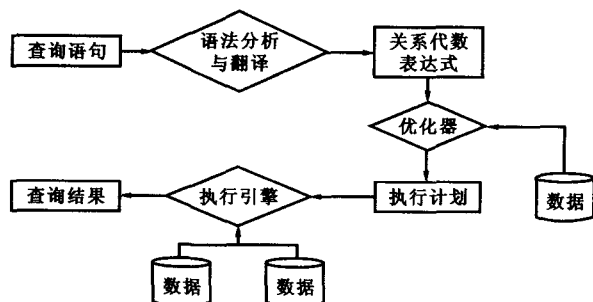


图5 查询的处理过程

(2) 查询分解. 分解器将根据元数据管理模块对查询进行分解重写, 去除查询中没有使用到的引用, 同时记录各子查询之间的组合关系。

(3) 对查询树进行优化. 选择和投影运算尽量往树叶方向下移, 把笛卡尔积运算通过附加选择和投影运算转换为连接运算。

(4) 形成查询执行策略. 说明需要访问哪些局部数据库系统, 如何组装中间结果, 按什么顺序执行连接查询及在哪里执行全局处理。

(5) 局部查询. 按照查询策略把各子查询与相应的数据源 Wrapper 进行通信, 形成各数据源的局部查询。

(6) 查询重组. 最后各 Wrapper 将子查询结果转换后传给 Mediator 进行组装, 由 Mediator 把全局查询结果返回给查询用户。

6 结束语

本文在提出了一种新型的异构数据集成系统的基础上, 分析了数据的查询处理过程. 针对现有的数据集成系统未能很好地解决底层异构数据源的“即插即用”问题, 数据查询中的全局模式同各数据源模式的转换工作繁杂, 用户视图不能很好地面向用户. 本文提出的数据集成系统以及在该系统中定义的查询有两个优点: (1) 查询过程是基于底层各异构数据

源之上的一层 Web Service, 只要可以提供 Web Service, 各异构数据源可以快速方便的查询; (2) 在数据查询中, 提出了全局可通用的数据模型, 使得各局部模式和全局模式基于同一个数据模型, 转化方便; (3) 用户的查询是基于快速生成的用户视图的, 使得用户能够方便、快速、直观地进行数据查询. 但是, 在集成数据的查询优化问题上, 本文还没有得到很好的解决, 在接下来的工作中, 将针对集成数据和本系统的特点, 提出一个更适合的查询优化方法。

参考文献:

- [1] 李军怀, 周明全, 耿国华, 等. XML 在异构数据集成中的应用研究[J]. 计算机应用, 2002, 22(9): 10-12.
- [2] 王宗杰, 侯贵法, 王成耀, 等. 基于元数据的分布异构数据集成研究[J]. 微计算机信息, 2007, 23(27): 84.
- [3] 于秀霞. 分布式数据查询优化算法的研究[J]. 长春大学学报, 2006, 16(1): 26-29.
- [4] 孙友仓, 宋彩利, 李润州. 一种基于 Web Service 的异构数据集成中间件[J]. 西安科技大学学报, 2007, 27(3): 472-474.
- [5] Sanjay Madria, Kalpdrum Passi, Sourav Bhowmick. An XML schema integration and query mechanism system[J]. Data & Knowledge Engineering, 2008(65): 266-303.
- [6] Boag S, Chamberlin D, Fernandez M F, et al. XQuery 1.0: an XML query language, W3C working draft [EB/OL]. (2002-11-15) [2009-12-19]. <http://www.w3c.org/TR/xquery/>.
- [7] Haiwei Ye, Kerherve B, Bochmann G V, et al. Pushing quality of service information and requirements into global query optimization[J]. Database Engineering and Applications Symposium, 2003(7): 170-179.

作者简介:

唐智勇 男, (1976-), 博士研究生. 研究方向为 web 服务与数据集成。

(上接第 35 页)

- [6] 杨晓伟, 廖光灯, 胡越明. 基于 Xen 的 X86 虚拟机性能调优[J]. 计算机工程, 2006, 32(24): 251-253.
- [7] Nurmi D, Wolski R, Grzegorzczak C, et al. The eucalyptus open-source cloud-computing system[C]// Cloud Computing and Applications 2008(CCA08). Germany: Hage, 2008.
- [8] Rajkumar Buyya. 高性能集群计算、编程与应用[M]. 郑伟民, 译. 北京: 电子工业出版社, 2001.
- [9] Alcaus Bayucan, Robert L Henderson. Portable batch system[M]. USA: Symposium on Cluster Computing and

the Grid, 2001.

作者简介:

曾龙海 男, (1987-), 硕士. 研究方向为人工智能与云计算。

张博锋 男, (1968-), 研究员. 研究方向为智能人机交互、云计算。

张丽华 女, (1982-), 硕士. 研究方向为人工智能与云计算。