

基于 HGAV 的多源异构数据集成方法^①

郑奎奎, 刘海滨

(中国航天系统科学与工程研究院, 北京 100048)

通讯作者: 郑奎奎, E-mail: jia_kuikui@163.com

摘 要: 针对信息系统中海量数据多源异构和难以共享的问题, 提出了多源异构数据虚拟集成框架. 数据集成系统中的 GAV (Global-As-View) 模式映射方法面对信息量分布不均匀的数据源时, 查询效率较低, 在对 GAV 改进的基础上, 提出了基于 HGAV (Hierarchical-Global-As-view) 的模式映射算法, 通过引入中间数据源模式, 形成分层的全局视图, 大大缩减了映射空间, 简化了映射集合, 便于查询的重写和优化. 利用宁东智慧环保项目中的五大类数据对本文所提出的算法加以验证, 实验结果表明该算法相较于 GAV 模式映射算法提高了数据集成效率, 缩短了查询时间.

关键词: 数据集成; 映射; 查询; HGAV; 中介模式

引用格式: 郑奎奎, 刘海滨. 基于 HGAV 的多源异构数据集成方法. 计算机系统应用, 2018, 27(3): 27-35. <http://www.c-s-a.org.cn/1003-3254/6235.html>

Multi-Source Heterogeneous Data Integration Method Based on HGAV

JIA Kui-Kui, LIU Hai-Bin

(China Aerospace Academy of Systems Science and Engineering, Beijing 100048, China)

Abstract: In order to solve the problem of heterogeneous data sources and data sharing in information systems, a virtual integration framework of multi-source heterogeneous data is proposed. Since GAV (Global-As-View) pattern mapping method in data integration system is less efficient when faced with uneven distribution of information, GAV method is improved, and the pattern mapping method based on HGAV (Hierarchical-Global-As-View) is proposed. By introducing the intermediate data source pattern, a hierarchical global view is formed, which greatly reduces the mapping space. In this way, the mapping set is simplified, and the query is easier to rewrite and optimize. The proposed algorithm is verified by the five main types of data in the Ningdong Intelligent Environment Protection project. The experimental results show that the pattern mapping algorithm based on HGAV improves the efficiency of data integration and shortens the query time, compared to the GAV schema mapping algorithm.

Key words: data integration; mapping; query; HGAV; mediation model

随着计算机及网络技术的迅猛发展和广泛应用, 政府和企业的信息化程度得到了大幅度的提高, 数据的采集、存储、处理和传播的数量也与日俱增. 数据不断累积, 形成了海量数据, 如果能够实现这些数据的共享, 将会使更多的人能更充分地使用已有的数据资源, 减少重复的数据收集等劳动和相应的费用. 然而,

这些海量数据往往存储在不同的平台和系统中, 数据具有异构多源的特征, 需要通过不同的方式进行访问, 难以实现数据的共享, 各个数据源也就变成了“信息孤岛”. 因此, 为实现不同数据源的互联互通和数据的共享, 构建多源异构数据集成系统显得尤为迫切.

在进行多源异构数据集成时, 有两种集成框架^[1],

^① 基金项目: 国家自然科学基金 (U150120175)

收稿时间: 2017-06-05; 修改时间: 2017-06-17; 采用时间: 2017-06-30; csa 在线出版时间: 2018-01-25

一种是通过建立数据仓库进行数据集成,一种是虚拟数据集成系统.数据仓库的方法需要将各个数据源的数据抽取转换装载到一个集中的数据库中,当数据源有更新时,数据仓库不能够及时随着数据源的更新而更新,实时性较差,而且建立数据仓库需要较大的经费成本.虚拟数据集成系统^[2]通过中介模式的方法,建立数据源模式与中介模式的映射关系,用户能够通过一张统一的视图实现对数据源的实时访问,无需再将各个数据源抽取转换装载到一个物理存储空间中,用户发起的查询请求会通过中间模式与数据源模式的映射关系将查询分解到各个数据源中,最后将各个数据源上的查询结果汇总就得到了完整的查询结果.考虑到数据集成系统实时性和建设成本的要求,本文采用虚拟数据集成的框架.

数据集成系统中核心的问题就是如何建立数据源模式与中介模式的映射关系^[2],只有解决了这个问题才能够实现多个数据源的数据集成.由此学者提出了模式映射语言^[3],以便于建立数据源模式与中介模式的映射关系.主要有三类模式映射:全局视图 GAV (Global-As-View)、局部视图 LAV(Local-As-View) 和全局-局部视图 GLAV (Global-and-Local-As-View). Glenn I N 在文献^[4]中提出了 ISTAR 数据融合框架,将情报、监视、目标捕获和侦察数据融合到统一的数据库中,该框架基于北约国家 IEEE POSIX 共同的数据规范,在数据的互操作基础之上实现多源异构信息的融合. Fonseca 在文献^[5]中提出了利用本体进行语义转换,建立同一数据集内概念的相互联系及不同数据集语义间的对应关系.以上方法能够解决多源异构数据集成的问题,然而在实际操作中集成效率和查询响应时间存在不足,尤其是当数据源的规模相差较大时就会出现效率低下的情况,即一系列需要集成的数据源中,有某个数据源包含信息量较大,并且与其他数据源有较强的关联关系时,例如宁东智慧环保项目中,需要实现环保类数据、企业投入产出类数据、气象数据、遥感卫星数据、地理信息系统数据的数据集成,且地理信息系统具有较大规模数据量,这种情况下进行多源异构数据集成的复杂度较高,传统的数据集成方法会使映射关系非常复杂,由此本文提出了一种层次全局视图 HGAV (Hierarchical-Global-As-View) 模式映射来提高数据集成效率和缩短数据集成系统的查询响应时间.

1 多源异构数据集成框架

虚拟数据集成系统面向用户查询的是具有统一视图的中介模式,数据源通过映射关系与中介模式建立对应关系.用户在中介模式上的查询会通过中介模式 G 与数据源模式 S 的映射关系 M,将查询分发到各个数据源上,最后再通过包装器将各个数据源查询得到的结果合并包装返回给用户.逻辑结构图如图 1 所示.通过系统集成,用户构建自己的查询时不必知道数据在哪里,以及源数据是如何组织的.数据源提供实际数据是需要集成的对象,它们独立存在于不同主机节点上,它可以是各种类型的数据库、数据文件等等.包装器位于整个系统的底层,驻留在数据源上.与数据源直接打交道.不同类型数据源需要不同的适配器.包装器主要功能有: (1) 请求数据源模式信息; (2) 响应数据源中心的数据提取请求,并对结果进行整理; (3) 将结果整理成 XML 格式,发送给集成中心.映射配置器建立全局模式与局部模式间的映射关系.

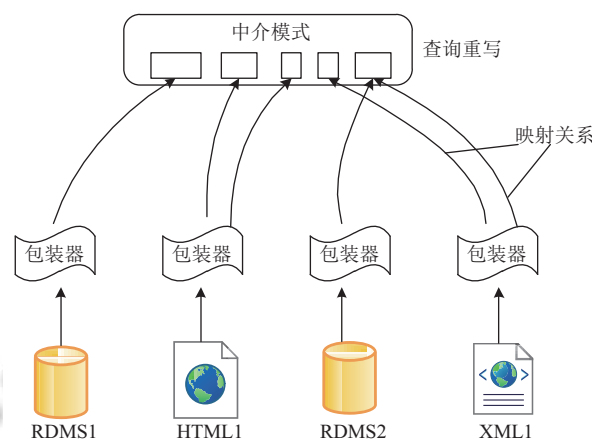


图1 虚拟数据集成框架

在整个虚拟数据集成系统中的查询包括查询重写、查询优化、查询执行等环节.一个中介模式上的查询经过查询重写模块后,会生成一个基于数据源的逻辑查询计划,该逻辑计划经过查询优化器后生成基于数据源的物理查询计划.该物理查询计划在执行引擎里执行生成各个数据源上的子查询,并根据计划的实施情况,将重优化请求发送给查询优化器.

在全局视图上的查询是针对 XML 数据查询的,因为本文所建立的全局视图是将局部数据源的模式转化为 XML Schema 形式,然后再将各个局部数据源的模式合并形成一张总的基于 XML Schema 的全局视图,

所以全局视图上的查询就是将 XML Schema 作为元数据, 利用 XQuery 作为查询语句在虚拟的集成数据库上查询。

中介模式中的术语组成了用户的查询语句。系统首先应把用户的查询语句重写成数据源模式所对应的查询语句。用户在中介模式上的查询被重写为数据源上的查询语句, 各个数据源的查询结果汇总起来就形成了用户所期望的查询结果。这个重写的结果就被称为逻辑查询计划^[6]。

利用 Datalog 表达形式来表示查询, 其形式如下:

$$q(X) :- p_1(Y_1), \dots, p_n(Y_n)$$

这里, q 和 $p_1, \dots, p_n$ 是谓词, q 代表查询结果, $p_1, \dots, p_n$ 指向数据源中的关系。 $q(X)$ 称为查询头, 其余部分称为查询体, $p_1(Y_1), \dots, p_n(Y_n)$ 称为查询体中的子查询。元组 $X, Y_1, \dots, Y_n$ 中仅包含变量或常量。如果 $X \subseteq Y_1 \cup \dots \cup Y_n$, 即查询体中所有的变量必须包含查询头中的所有变量, 则认为这样的查询结果是完备的。带有比较谓词($<, \leq, =, >, \geq$)的子查询也可以包含在查询体中, 前提条件是如果子目标中含有比较谓词, 且 x 是比较中的一个变量, 那么在普通子目标中也必须包含它。我们通常用 $Q(D)$ 表示在数据库实例 D 上的查询结果。

对于任意数据库实例 D , 如果查询 Q_1 在此数据库上计算的结果都包含于另一个查询 Q_2 的结果, 则记为 $Q_1(D) \subseteq Q_2(D)$, 即 Q_1 包含于 Q_2 , 记作 $Q_1 \subseteq Q_2$; 如果 $Q_1 \subseteq Q_2$ 且 $Q_2 \subseteq Q_1$, 则称 Q_1 和 Q_2 是等价的, 记作 $Q_1 = Q_2$ 。

在多源异构数据集成系统中, 最核心的就是对多个数据源进行模式匹配和模式映射, 只有建立一套完整的语义匹配和语义映射系统, 用户才能通过一张统一的全局视图访问到底层的数据源数据。本文在现有的 GAV 模式映射的基础上, 提出了分层的 HGAV 模式映射方法, 接下来将要介绍 HGAV 模式映射方法的定义, 以及模式匹配和模式映射的方法。

2 基于 HGAV 的模式映射

2.1 HGAV 模式映射的定义

在虚拟数据集成系统中, 核心的部分就是建立局部数据源与全局视图的映射关系, 这个映射关系高效与否直接决定了整个数据集成系统的效率高低, 因为用户针对中介模式提出的查询可能涉及到多个数据源,

每个数据源都有一个对应的数据源描述, 模式映射就是将各个数据源的模式映射到全局模式中, 并且映射后的模式互补, 不发生查询冲突。常用的映射方法是 GAV 模式映射, 然而 GAV 的映射方法面对信息量分布不均匀的多个数据源时, 查询效率较低, 因此本文提出了改进方法, 利用基于 HGAV 的模式映射方法优化查询, 从而提高数据集成系统的查询效率, 缩短查询的响应时间。

1) 模式映射语言

模式映射是描述模式之间关系的表达式集合^[7]。这里的模式映射描述了中介模式和源模式之间的关系。当一个查询针对中介模式查询时, 我们会使用这些映射把它重写为对数据源的查询。重写的结果是一个逻辑查询计划。我们使用查询表达式作为模式映射的主要表达方式, 在描述中, 用 G 表示中介模式, 用 $S_1, \dots, S_n$ 表示源模式。

2) 模式映射语义

模式映射的语义是这样指定的, 通过定义中介模式的哪个实例与数据源的给定实例是一致的^[8]。具体来讲, 一个语义映射 M 定义了一个关系 M_r :

$$I(G) \times I(S_1) \times \dots \times I(S_n)$$

其中, $I(G)$ 表示中介模式的可能实例, $I(S_1), \dots, I(S_n)$ 表示相应源关系 $S_1, \dots, S_n$ 的可能实例。如果 $(g, s_1, \dots, s_n) \in M_r$, 那么当源关系实例是 $s_1, \dots, s_n$, g 就是中介模式的可能实例。中介模式上查询的语义就是基于关系 M_r 的查询。

已知 GAV 模式映射的定义为^[9]: 假定 G 是一个中介模式, $S = \{S_1, \dots, S_n\}$ 是 n 个数据源模式。一个全局视图模式映射 M 是一个满足 $\bar{X} \supseteq Q(\bar{S})$ 或者 $\bar{X} = Q(\bar{S})$ 的表达式集合, 其中: (1) $\bar{X}$ 是中介模式 G 的一个关系, 并且最多出现在 M 的一个表达式中; (2) $Q(\bar{S})$ 是 $\bar{S}$ 中关系上的一个查询。

HGAV 模式映射相较于 GAV 就是抽象出一层介于全局模式和数据源模式的中间数据源模式, 中间数据和其他局部数据源一样, 也是数据库实例, 只不过它包含的数据库信息较多, 基于全局视图的查询大部分会发生在中间数据源上, 除非有些数据必须要从其他局部数据源获得, 具体定义如下: 假定 G 是一个中介模式, $S = \{S_1, \dots, S_n\}$ 是 n 个数据源模式, $S_m = \{S_{m1}, \dots, S_{mk}\}$ 表示介于中介模式和数据源模式之间的中间数据源模式。首先建立中间数据源到全局视图的模式映射 M_1 ,

M_1 满足 $\bar{X} \supseteq Q(\bar{S}_m)$ 或者 $\bar{X} = Q(\bar{S}_m)$, 其中: (1) $\bar{X}$ 是 G 中的一个关系, 并且最多出现在 M_1 的一个表达式中; (2) $Q(\bar{S}_m)$ 是 $\bar{S}_m$ 中关系上的一个查询.

同时又要建立数据源到中间数据源的模式映射 M_2 , M_2 满足 $\bar{S}_m \supseteq Q(\bar{S})$ 或者 $\bar{S}_m = Q(\bar{S})$, 其中: (1) $\bar{S}_m$ 是中间数据源中的一个关系, 并且最多出现在 M_2 的一个表达式中; (2) $Q(\bar{S})$ 是 $\bar{S}$ 中关系上的一个查询. HGAV 模式映射结构图如图 2 所示.

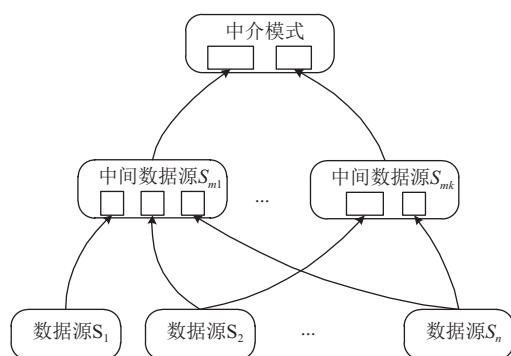


图2 HGAV 模式映射结构图

中介模式作为全局视图, 在中介模式上的查询通过映射 M_1 与中间数据源模式 S_m 产生对应关系, 任意中间数据源模式 S_{mi} 作为数据源 S 之上的“全局视图”(这里的全局视图也是局部的, 它是底层数据源上的全局视图). 通过映射 M_2 与 S 产生对应关系, 这样就将在中介模式上的查询分层级地映射到中间数据源模式和数据源模式上. 这样做有利于将包含信息量较大的数据源与一般数据源分离处理, 简化映射算法, 提高了数据集成的效率, 缩短了查询时间.

为了更形象地说明整个映射过程, 这里用几个数据表实例来说明, 已知两个学校 SchoolA 和 SchoolB, 它们都有图书馆图书信息表、图书借阅信息表、教师信息表和学生信息表, 具体字段如下所示:

SchoolA

Book(book_id, name, authors, type, ztflh)
Info(id, name, borrow_date, ztflh, return_date)
Teacher(id, name, gender, department, course)
Student(id, name, gender, department, class)

SchoolB

Book(book_id, title, authors, type, CLC)
Book_info(identity_id, name, bor_date, CLC, ret_date)
Teacher(id, name, gender, depart, major, course)

Student(id, name, gender, department, class)

这里要做一个学校 A 和学校 B 图书馆的数据集成, 两个学校的学生和老师既可以到学校 A 的图书馆借书也可以到学校 B 的图书馆借书. 那么需要将学校 A 和 B 的图书馆数据库、学生数据库和教师数据库中的数据集成, 按照本文所提出的 HGAV 模式映射方法, 由于图书馆数据库包含的信息量比较大, 所以将两个学校的图书馆数据库作为中间数据源, 将学生和教师数据库作为底层的局部数据源, 分别映射到图书馆数据源模式中, 最后再将两个图书馆数据源模式合成, 形成一张全局视图, 结果如下所示:

中间视图

LibraryA(book_id, book_name, authors, ztflh, borrow_date, borrow_id, borrow_name, department, return_date)

LibraryB(book_id, title, authors, CLC, bor_date, id, name, department, ret_date)

全局视图表

Library(book_id, title, authors, CLC, borrow_id, borrow_name, depart, return_date)

以上阐述了 HGAV 模式映射的定义, 以及利用实例说明了 HGAV 映射结果, 接下来将详细描述模式匹配和模式映射的过程.

2.2 模式匹配

为了创建数据源的描述信息, 我们经常首先创建模式匹配, 然后从匹配得到映射^[10]. 之所以首先创建匹配, 是因为比较容易从设计者哪里得到匹配. 模式匹配的主要目的就是在给定的模式 S 和 T 之间产生一个匹配 (即, 对应) 集合, 通过计算数据字段之间的相似度来确定数据是否匹配, 如 $title \approx name$, $ztflh \approx CLC$, $depart \approx department$. 由于可用的线索和启发式比较多, 并且还需要对匹配的准确度进行最大化, 所以需要构架一个模式匹配系统架构, 如图 3 所示.

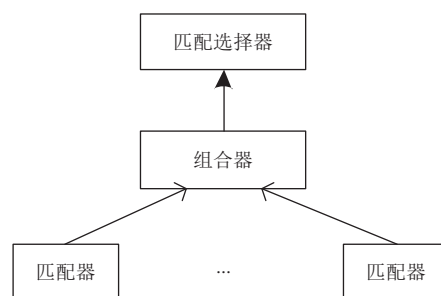


图3 模式匹配系统模块图

1) 匹配器

匹配器的输入是一对模式 S 和 T ^[11]. 除此之外, 匹配器还可以考虑任何其他可用的信息, 如数据实例、文本描述等. 匹配器输出一个相似度矩阵, 该矩阵为 S 和 T 中的每一个元素对 (s, t) 赋一个 0-1 之间的数值, 用来预测 s 是否与 t 匹配. 在具体应用中, 使用多种匹配器相结合的方法来获得相似度矩阵. 典型地, 有名字匹配器和数据实例匹配器两大类, 其中名字匹配器的度量算法有编辑距离、Jaccard 度量和 Soundex 度量算法, 实现实例匹配器的方法有 3 种: 创建识别器方法、测量度量值的重叠度方法和构建分类器的方法, 常用的分类器有朴素贝叶斯、决策树、规则学习和支持向量机等.

2) 组合器

不同的匹配器往往会计算出不同的相似度矩阵^[12], 这里就需要组合器将匹配器输出的多个相似度矩阵合并成一个. 简单的组合器可以取得分的均值、最小值或者最大值. 如果匹配系统采用 k 个匹配器来预测 s_i 和 t_j 的相似度得分, 那么均值组合器就可以采用如下的公式来计算两个元素之间的相似度得分:

$$\text{combiner}(i, j) = \left[\sum_{(m=1)}^k \text{matcherScore}(m, i, j) \right] / k$$

其中 $\text{matcherScore}(m, i, j)$ 是第 m 个匹配器输出的 s_i 和 t_j 的相似度得分.

3) 匹配选择器

匹配系统的最后一个模块就是从组合器输出的相似度矩阵中产生匹配^[13]. 最简单的匹配策略是阈值法: 相似度分数大于给定阈值的所有模式的元素对都可以作为匹配返回.

在对信息较为密集的多个数据源进行数据集成时, 利用阈值法能够解决数据的模式匹配问题, 因为面向同一个业务的系统中虽然存在多个数据源模式, 但是大都遵循着固定的行业标准, 所以进行模式匹配过程中, 相同的数据具有较高的相似度, 不同的数据具有较低的相似度, 利用阈值法能比较快捷有效地筛选出相同的模式. 当面对比较稀疏的数据模式, 且业务领域跨度比较大时, 简单的阈值法有可能失效, 需要采用更复杂的方法, 如基于贝叶斯的选择器、基于规则的选择器等.

2.3 模式映射算法

在把匹配转义为映射的过程中, 关键的挑战是匹配充实化、具体化^[14], 并把所有的匹配变成一个统一的整体在创建映射时, 需要通过连接和并操作对源和目标中数据的表组织结构进行调整, 使其一致^[15]. 本节主要描述如何探寻可能的模式映射空间. 给定一个匹配集合, 我们设计一个针对可能的映射的搜索算法, 这些可能的映射与给定匹配相一致. 我们根据一些常见的模式设计原理来定义合适的搜索空间.

在匹配集合已知的情况下, 本文设计了一个模式映射算法来实现数据源模式到全局视图模式的映射, 算法的输入是一个匹配集合: $M = \{f_i : (\overline{A_i} \approx B_i)\}$, 其中 $\overline{A_i}$ 是源 S 的属性集合, B_i 是目标 G 的一个属性.

输入: 中介模式 G 和数据源 S 模式之间的匹配, $M = \{f_i : (\overline{A_i} \approx B_i)\}$ (当 $\overline{A_i}$ 包含的属性个数大于 1 时, 形式为 $g(\overline{A_i})$, 其中 g 是对属性进行合并的函数); filter_i : 与 f_i 相关联的过滤器集合.

输出: 查询形式的映射.

{阶段 1: 创建候选集合}

设 $V := \{v \subseteq M | G \text{ 中的每个属性 } v \text{ 最多被提及一次}\}$

{阶段 2:}

$\Omega := V$

for 对于任何一个 $v \in V$ do

if $(A_i \approx B_i) \in v$, $\overline{A_i}$ 包含 S 中多个关系中的属性 then

使用启发式规则 1 和 2 来搜索连接 $\overline{A_i}$ 中的关系的连接路径

if 不存在连接路径 then

从 Ω 中返回 v

end if

end if

end for

{阶段 3:}

$\text{Covers} := \{\Gamma \subseteq \Omega, \Gamma \text{ 包含了 } M \text{ 中的所有匹配 } f_i, \text{ 并且 } \Gamma \text{ 的任何子集都无法包含所有的 } f_i\}$

$\text{selectedCover} := c \in \text{Covers}, c \text{ 是具有最少候选路径的覆盖}$

if Covers 中有多个覆盖 then

利用启发式 3 选择一个覆盖

end if

{阶段 4:}

for 对于 selectedCover 中的每一个覆盖 v do

创建一个如下形式的查询队 Q_v :

$\text{SELECT vars FROM } t_1, \dots, t_k \text{ WHERE } c^1, \dots, c^j, p_1, \dots, p_m$

其中, vars 表示 v 中匹配涉及的属性, $t_1, \dots, t_k$ 是 v 的连接路径中的各个关系, $c^1, \dots, c^j, p_1, \dots, p_m$ 是 V 的连接路径中的连接条件, $\text{filter}_1, \dots, \text{filter}_m$ 是连接路径中的过滤条件

end for

return 查询 $Q_1 \text{ UNION ALL } \dots \text{ UNION ALL } Q_b$

其中, $Q_1, \dots, Q_b$ 是前面创建的查询.

启发式 1. 寻找连接路径. 一个连接路径可以是:

1) 外键之间的路径;

2) 通过检查以前 S 上的查询而得到的路径;

3) 通过挖掘 S 中可连接的列的数据而发现的路径.

在 V 中, 我们需要为寻找连接路径的候选集合的集合记为 Ω . 如果存在多个连接路径, 那么使用如下启发式进行选择.

启发式 2. 选择连接路径. 优先选择外键之间的路径. 如果存在多个这样的路径, 那么选择在匹配中某一个属性上有过滤器的路径 (如果存在这样的路径). 为了对路径进一步排序, 选择外连接和内连接之间估计差别最小的连接路径. 最后可以选择拥有最少不确定元组的连接路径.

启发式 3. 选择覆盖. 如果存在多个覆盖, 那么选择候选集合数量最少的一个, 因为一般情况下我们认为, 越简单的映射越合适. 如果有多个覆盖含有相同数目的候选集合, 那么选择包含较多 MS 属性的覆盖.

这样就建立了中间数据源模式与局部数据源模式的映射关系, 然后再对全局模式与中间数据源按照以上算法建立映射关系, 如此就可以形成一个完整的 HGAV 模式映射系统了.

2.3 基于 HGAV 的模式映射过程

利用 2.2 节中的模式匹配方法实现不同数据源的语义匹配, 建立匹配集合. 不同的匹配器往往会得到不同的匹配相似矩阵, 需要根据字段所在的语义环境判断所匹配的对象, 图 4 展示了模式匹配的对应关系.

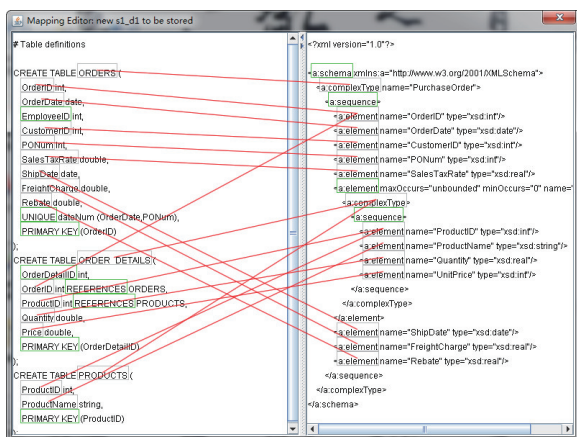


图 4 模式匹配对应关系

图 4 中实现了从关系数据库模式到 XML Schema 的转换, 按照字段的相似程度建立了匹配关系. 利用模型转换算法可以将所有的数据库模式定义信息

转换为 XML schema, 然后按照 HGAV 模式映射方法, 将底层的数据源模式映射到中间数据源模式, 再将中间数据源模式映射到全局视图模式中.

完成了数据的模式匹配之后, 就可以利用 2.3 节中的映射方法建立数据源上的查询了, 根据其中的三个启发式规则自动地实现模式映射的建立, 查询所用到的元数据信息从全局 XML schema 获得, 通常采用 XQuery 语句进行查询, 如:

```
for $x in doc("purchase.xml")/PurchaseOrder
where $x/Product/ProductName="juice"
order by $x/Product/ProductID
return $x/Product/ProductName
```

如果所做的查询涉及到多个数据库, 就需要进行查询的分解, 将查询分解成针对各个局部数据源的查询, 如果数据源是 MySQL、Oracle 等结构化的数据, 还需要将 XQuery 查询转换为相应的 sql 语句. 最后再局部数据源上的查询结构再通过 wrapper 包装器, 将查询结果包装成 XML 形式的数据文档返回结果, 最后将所有返回的 XML 文档合并就形成了总的查询结果了, 整个数据集成系统模型如图 5 所示.

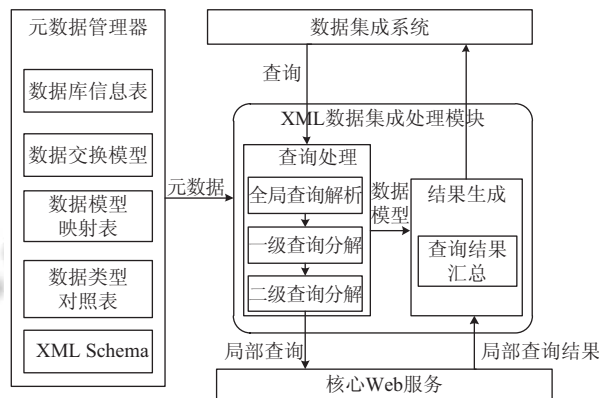


图 5 宁东智慧环保数据集成系统模型图

2.4 HGAV 算法的复杂度

假定 $\overline{M_1} = M_{11}, \dots, M_{1n}$ 是 G 和 $S_m = \{S_{m1}, \dots, S_{mk}\}$ 之间的模式映射, 同理 $\overline{M_2} = M_{21}, \dots, M_{2t}$ 是 G_{mi} 和 $S = \{S_1, \dots, S_n\}$ 之间的模式映射. 如果 Q_m 和 M_1 中的 Q_{mi} 是合取查询或者合取查询的并集, 那么即使有比较和否定原子操作, 找到 Q_m 的全部确定结果也是中间数据源大小的多项式时间问题 (PTIME), 同时重写的复杂度是查询和中间源描述大小的多项式时间问题. 同理如果 Q 和 M_2 中的

Q_i 是合取查询或者合取查询的并集, 那么即使有比较和否定原子操作, 找到 Q 的全部确定结果也是数据源大小的多项式时间问题, 同时重写的复杂度是查询和源描述大小的多项式时间问题. 假设中介模式和 S_m 之间的查询复杂度为 $O(k)$, 中间数据源模式 S_m 与数据源模式 S 的查询复杂度为 $O(t)$, 则总的查询复杂度为 $O(kt)$. 如果采用 GAV 或者 LAV 的模式映射方法, 查询复杂度为 $O((k+t)(k+t))$, $O(kt) < O((k+t)(k+t))$. 故采用 HGAV 模式映射的查询复杂度远远小于 GAV 和 LAV 模式映射的查询复杂度.

3 基于 HGAV 的宁东智慧环保数据集成系统

3.1 面向宁东智慧环保的数据集成系统体系架构

宁东智慧环保工程中包含很多类别的数据, 主要有环保类数据、企业投入产出数据、气象数据、遥感卫星数据以及 GIS 数据. 这些数据分散在不同的应用系统之内, 数据结构多样, 需要利用多源异构数据的集成方法将各类数据进行集成, 以便于进行查询和数据挖掘. 本文采用基于 HGAV 模式映射的方法, 通过建立中介模式在各类数据源上建立一张统一的视图, 用户就可以通过这个统一的视图进行查询. 整个宁东智慧环保工程多源异构数据集成系统的体系架构如图 6 所示. 数据集成系统需要从各个数据库的元数据中读取数据的结构、模型和映射关系, 进而按照本文所提出算法解析查询, 并将查询分解成针对各个数据源的局部查询, 最后获得各个局部查询的汇总结果.

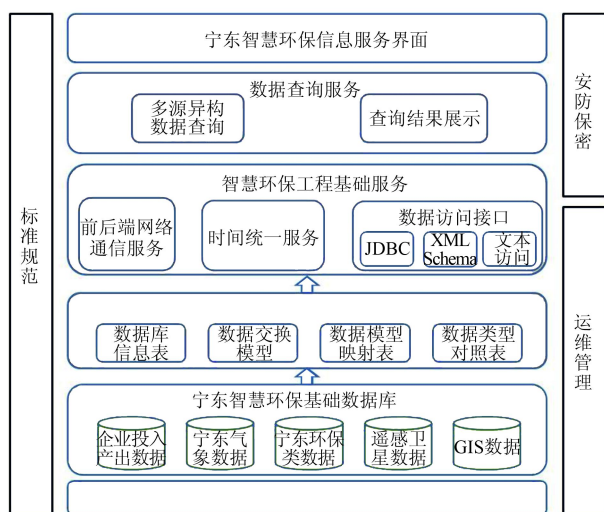


图6 宁东智慧环保工程多源异构数据集成系统体系架构图

在该虚拟集成系统中建立了两个模式映射集合, 包括中介模式与中间数据源模式之间的映射集合 M_1 , 以及中间数据源模式与底层数据源的映射集合 M_2 . 在智慧环保的数据集成系统中, 共有 5 个数据源, 分别是环保数据库、企业投入产出数据库、气象数据库、遥感数据库以及 GIS 数据库. 在这 5 类数据库中 GIS 数据库包含的信息量最大, 而且其他 4 个数据库与 GIS 数据存在地理位置的关系, 故把 GIS 数据源作为中间数据源, 其他 4 个数据源作为底层的数据源. 在 GIS 数据源之上建立中介模式, 是用户查询的接口.

3.2 实例验证

为了验证本文所提出方法的正确性, 设计了一个基于 Java 的 Web 应用系统, 通过浏览器客户端访问数据集成系统.

实验运行环境: 硬件配置为 Intel(R) Core(TM) i3 CPU M 350 @2.27 GHz; 4 GB RAM 内存; Windows 7(64 位) 旗舰版操作系统; Apache-tomcat-8.0.15 服务器环境. 软件设计是在 eclipse-4.4.1 平台下利用 Java 语言编写完成的.

利用 chrome 浏览器作为 Web 客户端, 用账户 user 登录系统, 可以看到宁东地区所有跟环保相关的数据, 包括污染源企业、污染源行业分布、废水在线监测信息、废气在线监测信息、空气质量、环境立体监测信息等. 智慧环保的多源异构数据融合系统效果图如图 7 所示. 通过利用所提出的多源异构数据融合模型将来自各个信息系统的数据融合, 形成了一个统一的视图, 用户通过这个统一的视图能够很便捷地访问到各类数据.

将所有排放污染物的企业位置 and 所有空气、水、固废监测站点与 GIS 相结合, 可以通过一张图清晰地看到所有企业的污染物排放量以及各个监测站点测得的环境污染物含量, 并能通过放置鼠标到地图上的任意一个点获得实时的环境信息, 效果如图 8 所示, 可以看到当前 PM2.5 含量为 23.765, PM10 含量为 73.047 等信息. 图中小图标表示该地点存在一个工厂, 可以清晰地看到工厂的分布位置, 同时在 GIS 图下方还能观察到当前排污的企业有哪些. 右侧是专家综合研讨区, 专家根据左侧图中的环境信息, 对某个相关议题发起讨论, 并对讨论结果进行表决. 由此可见, 多源异构数据融合系统能够为用户提供快捷的数据访问, 而且能够较为全面地看到各方面的数据, 可以决策人员提供全面的信息.



图7 多源异构环保数据融合结果图

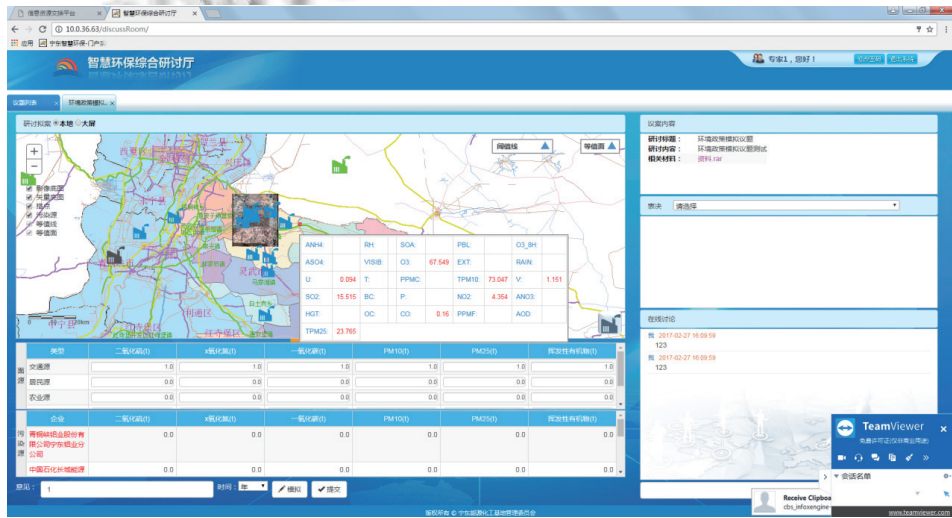


图8 GIS平台上的多源异构数据融合结果展示图

3.3 算法性能

为了验证本文所提出算法的正确性和高效性, 将其与常用的数据集成方法进行性能上的对比. 典型地, 这里与全局视图 GAV 和局部视图 LAV 模式映射方法进行对比. 使用的数据来自本地 3.1 中的五类数据, 这五类数据分别存储于 MySQL、Oracle、Access、SQL Server 和 XML 数据库中, 在同一个查询条件下, 分别记录在不同的数据量情况下, 这三种方法所整合的查询结果的准确率和计算时间, 如表 1 所示.

从表中可以看出, 三种算法的都有较高的准确率, HGAV 模式映射方法的准确率略高一些. 随着数据量的增大, 三种方法所消耗的时间都在增加, GAV 和

LAV 方法的时间消耗呈较快的增长趋势, 而 HGAV 方法所消耗的时间平稳增长. 由此可见, 本文所提出的 HGAV 模式映射方法高效, 且具有较高的准确率.

表 1 算法性能数据

数据条数(百万)	算法	准确率(%)	消耗时间(s)
1	HGAV	95	5.03
	GAV	93	5.31
	LAV	92	5.48
2	HGAV	94	6.01
	GAV	92	8.11
	LAV	90	9.18
3	HGAV	93	6.99
	GAV	91	11.02
	LAV	90	11.45

4 结论

本文构建了面向宁东智慧环保的多源异构数据集成系统,通过前台的数据请求,得到了经过集成处理后的数据信息,验证了基于HGAV的模式映射算法的可行性.为了验证该查询方法结果的完备性,与传统的GAV模式映射方法做了比较,发现查询的结果相同,由此可见基于HGAV的模式映射能够查询到正确的结果.最后通过将本文所提出的方法应用实际工程项目中,通过程序运行效果验证了本文所提出方法的可行性和实用性.

参考文献

- 1 Doan AH, Halevy A, Ives Z. Principles of Data Integration. Waltham, MA: Morgan Kaufmann, 2012. 110–120.
- 2 Lenzerini M. Data integration: A theoretical perspective. Proceedings of the Twenty-first ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems. Madison, WI, USA. 2002. 233–246.
- 3 Abiteboul S, Duschka O. Complexity of answering queries using materialized views. Proceedings of the 17th ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems. Seattle, WA, USA. 1998. 254–263.
- 4 Glenn IN. Multi-source data fusion in NATO coalition operations (a Canadian Army perspective on ISTAR). Proceedings of the Conference Record of the Thirty-Third Asilomar Conference on Signals, Systems and Computers. Pacific Grove, CA, USA. 1999. 407–411.
- 5 Fonseca FT, Egenhofer MJ, Agouris P, *et al.* Using ontologies for integrated geographic information systems. Transactions in GIS, 2002, 6(3): 231–257. [doi: [10.1111/1467-9671.00109](https://doi.org/10.1111/1467-9671.00109)]
- 6 Ullman JD. Information integration using logical views. Theoretical Computer Science, 2000, 239(2): 189–210. [doi: [10.1016/S0304-3975\(99\)00219-4](https://doi.org/10.1016/S0304-3975(99)00219-4)]
- 7 姚崇东. 基于XML的多源异构数据集成的实现方法研究[硕士学位论文]. 哈尔滨: 哈尔滨工程大学, 2007.
- 8 朱珊娜, 李书琴, 安福定. XML文档到关系数据库的转换研究. 计算机工程与设计, 2008, 29(21): 5507–5509, 5571.
- 9 张永新. 面向Web数据集成的数据融合问题研究[博士学位论文]. 济南: 山东大学, 2012.
- 10 许平格. 数据库管理系统中查询优化的设计和实现[硕士学位论文]. 杭州: 浙江大学, 2005.
- 11 钟将, 宋娟. 基于本体的异构数据集成框架. 计算机工程, 2011, 37(14): 44–46. [doi: [10.3969/j.issn.1000-3428.2011.14.013](https://doi.org/10.3969/j.issn.1000-3428.2011.14.013)]
- 12 刘伟, 孟小峰, 孟卫一. DeepWeb数据集成研究综述. 计算机学报, 2007, 30(9): 1475–1489.
- 13 化柏林. 多源信息融合方法研究. 情报理论与实践, 2013, 36(11): 16–19.
- 14 王艳华. 基于中间件技术的分布式数据集成研究与实现[硕士学位论文]. 武汉: 武汉理工大学, 2006.
- 15 Gao JJ, Xiao JQ. Research on heterogeneous data access and integration model based on OGSA-DAI. Proceedings of the 2013 5th International Conference on Computational and Information Sciences. Shiyang, China. 2013. 1690–1693.