基于模式映射的查询计划生成算法

李 由[1] 刘东波[2,3] 张维明[1]

(国防科技大学信息系统与管理学院 长沙 410073)[1]

(华中科技大学计算机科学与技术学院 武汉 430000)[2] (中国电子设备系统工程研究所 北京 100039)[3]

摘 要 因特网的迅速发展使得多数据源综合集成日益重要。但是,不同数据源之间数据结构和语义的异构性导致数据集成是相当困难的。本文提出了一种基于模式映射的查询计划生成算法。该算法在正确定义映射规则的前提下,根据不同的查询条件和不同的数据源模式,自动构造查询计划,并保证结果数据满足目标模式结构与引用完整性要求。

关键词 模式,映射,数据集成

A Query Discovery Algorithm Based on Schema Mapping

LI You[1] LIU Dong-Bo[2,3] ZHANG Wei-Ming[4]

(Department of Management Science & Engineering, National University of Defense Technology, Changsha 410073)[1]

(College of Computer Science & Technology, Huazhong University of Science & Technology, Wuhan 430000)[2]

(Institute of China Electronic System Engineering, Beijing 100039)[3]

Abstract With the rapid development of Internet, integration of diverse data sources becomes a more and more important issue. However, it is a difficult problem due to heterogeneous data structure and semantics of these data sources. This paper introduces a query discovery algorithm based on schema mapping. It can automatically construct query plans according to the target queries and data source schema by exploiting the mappings among data sources. It also guarantees that the target result satisfies the target structure and reference constraints.

Keywords Schema, Mapping, Data integration

1 引言

随着数据存储和网络技术的迅速发展,数据源的规模和数量日益增大。这些数据源大都具有分布、异构、自治和动态等特点,形成了大量的"数据烟囱",限制了系统之间的互操作。人工方式的数据集成工作费时、费力且容易出错,随着数据源数量的增多和规模的增大,这种方式甚至是不切实际的[5]。如何快速、有效、自动地实现异构数据源的集成,是解决互操作问题的关键。

当前,模式集成和模式映射是最主要的两种异构数据集成方法[2,4]。传统的异构数据源集成主要采用自底向上的模式集成手段,它根据多个数据源(本地)模式生成目标(全局)模式,即建立多个本地数据源模式的全局视图。全局模式依赖于本地模式而存在,不具有独立的语义信息。模式映射方法是一种自顶向下的方法,它主要通过建立模式之间的映射关系,完成异构数据源之间的数据转换,而每个数据源模式有独立的语义和结构。与模式集成相比,这种方法更灵活、更符合实际应用的需要、更具可操作性[7]。

本文提出了一种基于模式映射的查询计划生成算法QDA(Query Discovery Algorithm)。该算法在正确定义映射规则的前提下,根据不同的查询条件和不同的数据源模式,自动构造查询计划,并保证结果数据满足目标模式结构和引用完整性要求。

2 查询计划生成算法 QDA

根据映射规则,查询计划生成算法可处理来自目标模式的查询请求,生成数据源可执行的查询计划。为了简化问题描述,我们以关系模型为例。

2.1 标记方法

- 符号 S 表示数据源模式。
- 符号 T 表示目标模式。
- Q^S 表示基于数据源模式 S 的查询请求,其查询结果记为 R^S。
- Q^T 表示基于目标模式 T 的查询请求,其查询结果记为 R^T。
- $A^S=\{s_1,s_2,\cdots,s_i,\cdots,s_p\}$ 表示 S 的属性集合,其中 s_i 表示属性,域值记为 $dom(s_i)$。
- $A^T=\{t_1,t_2,\cdots,t_i,\cdots,t_q\}$ 表示 T 的属性集合,其中 t_i 表示属性,域值记为 $dom(t_i)$。

定义 1 令 S 表示数据源模式,T 表示目标模式,映射规则 m_i 定义了模式元素之间的关联关系,形式化表示为:

$$dom(s_i)\times\cdots\times dom(s_j)\cdots\times dom(s_k)\rightarrow dom(t)\cup\{null\},\quad (s_j\in A^S, t\in A^T)$$

映射集合为映射规则的集合,记为 M。

定义 2 数据源模式 S 可表示为一个置标的有向图 $G=\langle V,E\rangle$,其中顶点集 V 是 S 中所有表节点构成的集合,E 是有

李 由 博士研究生,主要研究方向为信息管理;刘东波 高级工程师,主要研究方向为数据工程;张维明 博士生导师,主要研究方向为信息管理。

向边 e 的集合。

定义 3 有向图 G 的每一条边 e 是有向置标的，它可以表示为一个三元组 $e=\langle v_i, v_j, Key\_cons\rangle$。$e$ 根据表元素的主外键关系建立：$\forall v_i, v_j \in V$。若表节点 v_i 中含有表节点 v_j 的外键，则建立有向边 $\langle v_i, v_j\rangle$。Key_cons 是边 e 上的标记，用于表示 v_i 与 v_j 之间的引用完整性约束条件。

利用关系演算形式化语言可将目标查询表示为：

$$\{\langle x_1, x_2, \cdots, x_u\rangle \mid P_T(x_1, x_2, \cdots, x_u)\}$$

其中，$x_1, x_2, \cdots, x_n$ 表示域变量，P_T 表示由原子构成的公式。

定义 4 查询 Q^T 的相关映射集合 $M^Q=\{m_i \mid TargetAttrs(m_i)\in\{x_1, x_2, \cdots, x_n\}\}$。其中，$TargetAttrs(m_i)$ 表示映射规则 m_i 的目标属性集合。

2.2 算法设计思想

定义 5 映射的基数表示通过一个联系集同另一实体相联系的实体数目[10]。根据映射的基数可将映射分为 1∶1、1∶n、n∶1 和 n∶m 四种基本类型。

模式的异构性导致了映射的多样性，QDA 算法对上述四种基本映射类型分别处理如下：

1)1∶1 类型映射。一方面根据映射规则集合 M^Q 对 Q^T 进行属性替换。另一方面，由于映射规则仅定义了属性级的关联关系，为了保证结果数据满足目标模式引用完整性要求，算法通过增加查询连接约束条件对各属性元组进行垂直方向上的合成，最终生成查询计划 Q^S。

2)n∶1 类型映射。通过水平分组算法将映射集合 M^Q 水平分解为若干个满足一定条件的、仅包含 1∶1 类型映射的子集合 M_i，将 n∶1 映射转化为 1∶1 映射，生成分组查询计划 Q_i^S，最终合并分组查询计划：$Q^S=\bigcup Q_i^S$。

3)1∶n 类型映射。按照定义 1 的映射规则表示方法，将映射分解为多个 1∶1 类型映射规则，并按照 1∶1 类型映射方法处理。

4)n∶m 类型映射。按照定义 1 的映射规则表示方法，将映射分解为多个 n∶1 类型映射，并按照 n∶1 类型映射处理方法处理。

2.3 QDA 算法描述

QDA 查询处理算法包括用于映射集合水平分组的 Grouping 子算法和用于计算连接约束条件的 Joining 子算法。

算法 1 查询计划生成主算法

输入：目标查询 Q^T；相关映射规则集合 M^Q.

输出：查询计划 Q^S.

算法开始：

```
/*构造模式 S 的有向图 G*/
G=(V,E)=Digraph(S);
/* 映射集合水平分组 */
Grouping(MQ,G)→{M1,M2,…Mk};
For each Mi
/* 计算连接约束条件 */
  Ji=Joining(Mi,G);
/*属性替换,增加 Ji,生成分组查询计划 */
  QiS=Replace(QT,Mi)+ Ji;
/* 合并分组查询计划 */
  QS=QiS;
算法结束.
```

水平分组阶段着重解决映射集合 M^Q 的水平分解问题。将 M^Q 分解为若干个满足一定条件的、包含多个一对一映射规则的子集合 M_i。分解后的 M_i 需满足以下条件：

1) $\forall m_l, m_k \in M_i (m_l \neq m_k)$，

$TargetAttrs(m_l) \cap TargetAttrs(m_k)=\Phi$；

2) $\forall t \in Attrs(Q^T)$，$\exists m_k \in M_i$ s. t. $t \in TargerAttrs(m_k)$；

3) $\forall v_l, v_k \in V_{sub}^i \rightarrow$ path. $Exis(v_l, v_k, G_0)=True$.

其中，$Attrs(Q^T)$ 表示 Q^T 的查询属性集合。条件(1)、(2)表明 Q^T 中的每个目标属性对应且仅对应一条映射规则。集合 $V_{sub}^i \subset V(G)$ 包含了所有 M_i 的数据源属性所在的表节点，G_0 为有向图 G 的基础图。条件(3)要求 G_0 中任意两个节点都是连通的，即表节点集合 V_{Sub}^i 中任意两个元素都有直接或间接的引用完整性关系。

水平分组子算法描述如下：

算法 2 Grouping 子算法。

输入：相关映射规则集合 M^Q；数据源模式 S 的有向图 G.

输出：映射分组集合 $M_{group}=\{M_k, \mid k=1, \cdots n\}$.

算法开始：

步骤 1 目标属性分组.

$A=TargetAttrs(M)$；

$A^\alpha=\{t \mid Num0fMapping(t)=1\}$；

/* 函数 NumOfMapping 用于计算目标属性 t 对应的映射规则数目 */

$A^\beta=A-A^\alpha$；

步骤 2 映射规则分组.

$M^\alpha=\{m_i \mid m_i \in M, TargetAttrs(m_i) \subset A^\alpha\}$；

For each $t \in A^\beta$

$M_t^\beta=\{m_i \mid m_i \in M, TargetAttrs(m_i)=t\}$；

步骤 3 映射集合重构. 从每个 M_t^β 中分别抽取映射规则，构建集合 M_i^γ，$(i=1,2,\cdots,\prod \mid M_t^\beta \mid)$，计算 $M_i=M_i^\gamma \cup M^\alpha$，重构后的 M_i 满足条件(1)(2)

步骤 4 /* 构建有向图 G 的基础图 G_0 */

$G_0=Underlying\_Graph(G)$；

步骤 5 筛选 M_i；

For each M_i

$V_{Sub}^i=SourceTables(M_i)$；

If for all $v_j, v_k \in V_{Sub}^i$ are connected in G_0

Then $M_{group}=M_{group} \cup \{M_i\}$；

/* M_i 满足条件(1)(2)(3) */

算法结束.

Joining 子算法计算数据源模式中各属性的连接约束条件 J_i，完成属性元组垂直方向上的合成。

算法描述如下：

算法 3 Joinning 子算法.

输入：分组后的映射规则集合 M_i；

数据源模式 S 的有向图 G.

输出：连接约束条件 J_i.

算法开始：

步骤 1 构建有向导出子图 H；

$V_{Sub}^i=SourceTables(M_i)$；

$V_0=Vertex(Path^{All})$；/* $Path^{All}$ 表示 V_{Sub}^i 中各顶点之间存在的所有有向路径 */

/* 构建由 V_0 导出的 G 的有向子图 H */

$H=G(V_0)$；

步骤 2 构建集合 V_{root}、V_{other}；

For H

$V_{root}=\{v_i \mid id(v_i)=0, v_i \in V_{Sub}\}$；/* 入度为零的根节点集合 */

$V_{other}=V_{Sub}^i - V_{other}$；

步骤 3 构建边集合 E'；

For each $v_j \in V_{root}$, each $v_k \in V_{other}$

/* 计算 $Path^{All}$ 中 v_j 到 v_k 的最短路径 */

$Path^S=ShortPath(v_j, v_k, Path^{All})$；

/* 提取 $Path^S$ 中所有有向边 */

$E_{path}=GetEdges(Path^S)$；

$E'=E' \cup E_{path}$；

步骤 4　构建连接约束条件集合 J_i；

For each $e_j \in E'$

/* 提取 e_j 的引用完整性约束条件 */ Key_cons = Get_$Key_cons(e_j, G)$; /* 构建连接约束集合 J_i */ $J_i = J_i \cup \{Key\_cons\}$;

算法结束.

定理 1　数据源 S 的某个数据实例 A，如果 A 满足 Q^S 的约束条件 P_S，则 A 必然也满足 Q^T 的约束条件 P_T。

证明：首先将 P_S 改写为析取范式。由算法 QDA 的约束条件 P_S 生成过程可知，P_T 的每个子合取式在 P_S 中都有对应的子合取式 P'_S。此外，P_S 中还包括 QDA 算法生成的连接约束条件 J_i，即 $P_s = P'_S \wedge J_i$。由此可知，P_T 条件不强于 P_S。由于 A 来源于 S 且满足 P_S，则 A 也必满足 P'_S 中的所有子合取式，而这个合取子式在 P_T 中必然有相应的子合取式，因此 A 必满足 P_T 中的相应子合取式，则 A 也满足 P_T。

证毕。

3　算法举例

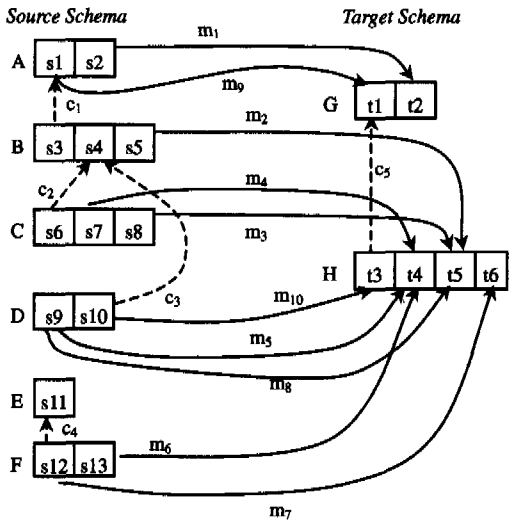

图 1　数据源模式与目标模式示例

例 3.1　QDA 算法举例。数据源模式和目标模式如图 1 所示，映射规则集合 $M=\{m_i \mid i=1,\cdots,10\}$，数据源模式中的约束关系 $C^S=\{c_1, c_2, c_3, c_4, c_5\}$。目标查询要求如下：

Q^T：SELECT $G.t2, H.t4, H.t5$ FROM G, H WHERE $(G.t1=H.t3)$ AND $(G.t2>3)$

相关映射规则集合为 $M^Q=\{m_1, m_2, m_3, m_4, m_5, m_6, m_8, m_9, m_{10}\}$。构建有向图 G 如图 2 所示。分组运算将 M^Q 分解为 M_1、M_2、M_3、M_4、M_5、M_6 映射子集合。经过 Joining 算法计算得 Q_i^S 值，如表 1 所示。

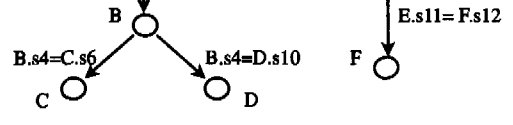

图 2　例 3.1 中的有向图 G

例 3.2　Grouping 算法举例。例 3.1 中，输入 M^Q 和有向图 G。

目标属性分组阶段：

$A^\alpha=\{t1, t2, t3\}$，$A^\beta=\{t4, t5\}$

映射规则分组阶段：

$M^\alpha=\{m_1, m_9, m_{10}\}$，$M_{t4}{}^\beta=\{m_4, m_5, m_6\}$，

$M_{t5}{}^\beta=\{m_2, m_3, m_8\}$。

映射集合重构阶段：

$M_1{}^\gamma=\{m_4, m_2\}$，$M_2{}^\gamma=\{m_4, m_3\}$，$M_3{}^\gamma=\{m_5, m_2\}$，

$M_4{}^\gamma=\{m_5, m_3\}$，$M_5{}^\gamma=\{m_4, m_8\}$，$M_6{}^\gamma=\{m_5, m_8\}$，

$M_7{}^\gamma=\{m_6, m_2\}$，$M_8{}^\gamma=\{m_6, m_3\}$，$M_9{}^\gamma=\{m_6, m_8\}$。

表 1　Q_i^S 值

Q_i^S	SQL Query
$Q_1{}^S$	SELECT A. s2, C. s7, C. s8 FROM A, B, C WHERE (A. s1=B. s3) AND (C. s6=B. s4) AND (A. s1=D. s10) AND (A. s2>3)
$Q_2{}^S$	SELECT A. s2, C. s7, B. s5 FROM A, B, C WHERE (A. s1=B. s3) AND (C. s6=B. s4) AND (A. s1=D. s10) AND (A. s2>3)
$Q_3{}^S$	SELECT A. s2, D. s9, B. s5 FROM A, B, C WHERE (A. s1=B. s3) AND (D. s10=B. s4) AND (A. s1=D. s10) AND (A. s2>3)
$Q_4{}^S$	SELECT A. s2, D. s9, C. s8 FROM A, B, C WHERE (A. s1=B. s3) AND (C. s6=B. s4) AND (D. s10=B. s4) AND (A. s1=D. s10) AND (A. s2>3)
$Q_5{}^S$	SELECT A. s2, C. s7, D. s9 FROM A, B, C WHERE (A. s1=B. s3) AND (C. s6=B. s4) AND (D. s10=B. s4) AND (A. s1=D. s10) AND (A. s2>3)
$Q_6{}^S$	SELECT A. s2, D. s9, D. s9 FROM A, B, C WHERE (A. s1=B. s3) AND (C. s6=B. s4) AND (D. s10=B. s4) AND (A. s1=D. s10) AND (A. s2>3)

筛选 M_i 阶段：V_{Sub}^1、V_{Sub}^8、V_{Sub}^9 中节点 F 为孤立节点，因而排除映射规则集合 M_7、M_8、M_9。最终得出 $M_{group}=\{M_1, M_2, M_3, M_4, M_5, M_6\}$。

例 3.3　Joining 算法举例。图 3 所示的数据源模式和目标模式中，其中，

m_l：$payRate(HrRate) * WorksOn(Hrs) \rightarrow personnel(Sal)$

Q^T：SELECT *Project.ProjName, Personnel.Sal*

FROM *Personnel, Project*

WHERE *Personnel.ID=Project.EmpID.*

经分组阶段计算后，输入：

$M_{group}=\{M_1\}=\{\{m_1, m_3, m_4, m_5\}\}$。

Joinng 算法计算各中间变量如下。

$V_{sub}^1=\{Project, WorksOn, PayRate\}$;

$Path^A=\{WorksOn\text{-}Project, WorksOn\text{-}PayRate, WorksOn\text{-}Student\text{-}PayRate\}$;

$V_0=\{Project, WorksOn, PayRate, Student\}$; $V_{root}=\{WorksOn\}$;

$V_{other}=\{PayRate, Project, Student\}$;

$E'=\{(WorksOn, Project), (WorksOn, PayRate)\}$;

$J_1=\{Project.ProjName=WorksOn.Pro, PayRate.Rank=WorksOn.ProjRank\}$;

Q^S：SELECT *Project.ProjName*, *WorksOn.Hrs * PayRate.HrRate* FROM *Project, WorksOn, PayRate*

WHERE *Project.ProjName=WorksOn.Proj*

AND *PayRate.Rank=WorksOn.ProjRank*.

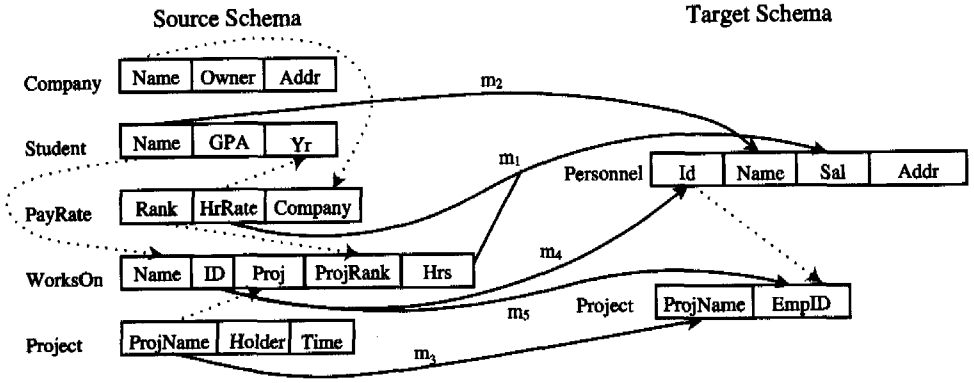

图 3　数据源模式与目标模式示例

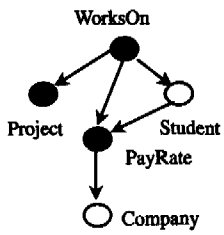

图 4　例 3.3 中的有向图 G

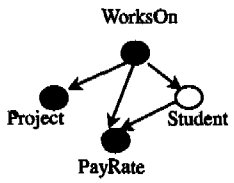

图 5　例 3.3 中的导出子图 H

结束语　模式集成和模式映射是当前异构数据源集成采用的两种主要方法。本文提出的基于模式映射的查询计划生成算法能够根据不同的查询条件和不同的数据源模式，自动构造查询计划，并保证结果数据满足目标模式结构与引用完整性要求，从而完成异构模式下的数据转换与集成。

参 考 文 献

1　Popa L, Velegrakis Y, Miller R J, et al. Translasting Web Data. In: Proc. VLDB, 2002. 598～609

2　Hernández M A, Miller R J, Haas L M. Clio: A Semi-Automatic Tool For Schema Mapping. SIGMOD, 2001

3　Fagin R, Kolaitis P G, Miller R J, et al. Data Exchange: Semantics and Query Answering. In: ICDT, 2003. 207～224

4　Miller R J, Haas L M, Hernhdez M. Schema Mapping as Query Discovery. In: Proc. of the Int'l Conf on Very Large Data Bases (VLDB), Cairo, Egypt, 2000. 77～88

5　Miller R J, Hernández M A, Haas L M, et al. The Clio Project: Managing Heterogeneity. SIGMOD Record, 2001, 30(1)

6　Domenigand R, Dittrich K R. An Overview and Classification of Mediated Query Systems. SIGMOD Record, 1999, 28(3)

7　Li You, Liu Dongbo, Zhang Weiming. A Data Tranformation Method Based On Schema Mapping. In: 3rd International Conference on Information Systems Technology and its Application, ISTA. Salt Lake City, USA 2004

8　李瑞轩，卢正鼎，等. 多数据库系统中基于 XIDM 的模式映射方法研究. 计算机研究与发展，2004，41(3)

9　陈彤兵，胡金化，等. 分布式自治数据源的联合查询. 计算机研究与发展，2004，41(4)

10　Silberschats A, Korth H F. 数据库系统概念. 杨冬青，等译. 北京机械工业出版社

(上接第 116 页)

(2)判断注册的资源的完备性：满足规则 2 的组合说明注册的具体服务资源是完备的，即组合的业务服务都可以匹配到至少一个具体服务。如果不满足规则 2，那么必须注册新的具体服务资源或修改服务组合以使得组合后的服务可执行。

本文从业务级服务组合的逻辑结构和注册的具体服务资源的完备性角度探讨了业务级服务组合的可执行能力，但影响业务级服务组合可执行能力的因素是多方面的。组合的服务之间不但存在控制流，也存在数据流，只有二者都正确流转才能真正保证组合的正常执行。本文对在组合阶段如何验证业务级服务组合的可执行能力进行了初步探讨，下一步的工作就在于研究组合中的数据流对业务级服务组合可执行能力的影响。

参 考 文 献

1　Han Y, et al. CAFISE: An Approach Enabling On-Demand Configuration of Service Grid Applications. Journal of Computer Science and Technology, 2003, 18(4)

2　van der Aalst, W M P. Interval timed coloured petri nets and their analysis. In Application and Theory of Petri Nets 1993, Proc. 14th International of Conference, Chicago, (USA), 1993, 691: 453～472

3　Morasca S, Pezzè M, Trubian M. Timed high-level nets. Journal of Real-Time Systems, 1991, 3: 165～189

4　van der Aalst W M P. The Application of Petri Nets to Workflow Management. The Journal of Circuits, Systems and Computers, 1998, 8(1): 21～66

5　李东来，韩燕波，王建武，喻坚. 面向服务应用中服务可用性及其引发的异常处理研究. 计算机研究与发展，2004，41(12)

6　Han Yanbo, Geng Hui, Li Houfu, et al. VINCA - A Visual and Personalized Business level Composition Language for Chaining Web-based Services. International Conference on Service Oriented Computing, Italy, 2003. 165～177

7　梁英，虎嵩林，李厚福，等. 面向网络化制造的网格应用平台及其核心技术研究. 计算机研究与发展，2004，41(12)

8　Fang Jun, Hu Songlin, Han Yanbo. A Service Interoperability Assessment Model for Service Composition. The Conference of SCC, 2004