

一种面向语义正确性的关系数据库访问方法*

唐富年,姚莉,李金洋

(国防科技大学 信息系统与管理学院,湖南 长沙 410073)

摘要:从异构关系数据源中获取语义正确的信息是当前利用数据资产辅助决策所面临的一大难题。为解决这一问题,提出了一种研究面向语义正确性的关系数据库访问方法(SCORDA)。SCORDA方法采用DL-Lite_{OWL}^N本体作为支持数据访问任务的概念视图,在本体的表达能力和推理复杂度之间做了一定折中;SCORDA方法通过LAV+GAV的映射方式建立本体与关系数据源之间的语义关联,兼顾了模式异构和阻抗失配问题;此外,SCORDA方法采用了一种动态ABox结构,即时从异构关系数据源中恢复虚拟对象,并且利用动态ABox进行实例检测推理任务,通过本体的模型语义来保证数据访问的正确性。通过原型系统验证了该方法的可行性。

关键词:本体;关系数据库;语义Web;数据语义

中图分类号:TP182 文献标志码:A 文章编号:1001-2486(2012)03-0103-06

A semantic correctness oriented relational database access method

TANG Funian YAO Li LI Jinyang

(College of Information System and Management, National University of Defense Technology, Changsha 410073, China)

Abstract: Nowadays, organization faces the problem of acquiring correct information from data asset to support decision-making. To solve this problem, a semantic correctness oriented relational data access method, namely SCORDA, is proposed. Firstly, ontology was used to describe the concept view, which is used in data access task, and description logic makes a suitable tradeoff between expressiveness and complexity. Secondly, a LAV+GAV mapping formalism was adopted to solve the schema heterogeneity and impedance mismatch problem. Furthermore, a special structure named dynamic ABox is provided to deal with the instance checking problem, and to ensure the semantic correctness of answers based on the model theory semantic of ontology. Finally, the feasibility of SCORDA method, through implementing a prototype system, was validated.

Key words: ontology; relational database; semantic Web; data semantic

数据是信息时代最重要的战略资源之一,关系数据库存储的数据是目前世界上最主要的数据来源^[1]。然而,尽管人类社会的数据资产逐年递增,但是如何从中访问到语义正确的信息,将信息优势转化为决策优势,仍然是当前很多组织面临的难题。

为解决上述问题,近年来基于本体的数据访问技术开始得到人们的关注^[2]。为了保证数据访问结果的语义正确性,通常需要从数据模型固有的内涵语义出发来认知和利用数据^[3]。其主要技术难点在于:

(1)由于关系模型和本体模型之间存在阻抗匹配问题^[4],必须采用较为复杂的语义映射,而这种映射的构造通常困难而繁琐。

(2)由于现有推理机只能支持中等规模

ABox的推理,传统的推理算法和推理机都难以直接应用于大规模的数据访问^[5]。

(3)在动态环境下,当本体或者关系数据库发生演化时,需要确保关系数据库和本体之间的映射保持正确,进而保证数据访问的语义正确性^[2]。

为此,本文提出了一种面向语义正确性的关系数据库访问(SCORDA, Semantic Correctness Oriented Relational Data Access,)方法。该方法采用DL-Lite_{OWL}^N本体作为支持语义访问的概念视图,通过LAV+GAV形式的映射方式建立关系数据库和本体之间的语义关联,借助动态ABox机制完成查询的处理。最后,通过原型系统和相关实验证明该方法是可行的。

* 收稿日期:2011-12-20

基金项目:国家自然科学基金资助项目(70971134)

作者简介:唐富年(1981-),男,内蒙古阿拉善盟左旗人,博士研究生,E-mail:tfn1981@gmail.com;

姚莉(通信作者),女,教授,博士,博士生导师,E-mail:Liyao6522@sina.com

1 基本框架与形式化描述

关系数据库的语义在概念建模阶段确定,而在数据库运行阶段,这些数据语义却经常难以被一般用户理解和重用。SCORDA 方法的基本思路是通过一个本体描述的概念视图将底层的异构数据源集成到本体知识表示模型的架构之下,借助知识库的知识表示和推理为上层应用提供语义正确的信息。采用 SCORDA 方法构建的信息系统可以如下定义。

定义 1 SCORDA 系统可定义为 $K = \langle E, O, DB, M \rangle$, 其中: E 为系统的论域, O 为一个能够充分描述关系数据库模式语义的本体, DB 是 E 上的关系数据库集合。 M 是关系数据库与本体之间的语义映射集合, $M = M_L \cup M_C$ 。其中, M_L 为关系数据库到本体的 LAV 映射, 而 M_C 为基于 M_L 构造的 GAV 映射。

本体的选择是 SCORDA 方法关键的环节, 直接决定了整个系统的性能。为了兼顾数据规模、本体表达能力和查询处理方面的需求, 本文在 DL-Lite 系列描述逻辑子集^[6]的基础上进行扩展。DL-Lite_A 描述逻辑子集是 DL-Lite 系列中面向数据访问设计的一个子集, 但是该子集牺牲了数量约束等对于关系数据库建模较为重要的约束。根据文献[7], 将某些 OWL 中的标准构造子加入 DL-Lite 子集后不会影响其计算特性, 例如属性的对称性和反自反性约束等; 在命名唯一假设(unique name assumption)和无角色包含的情况下, 引入数量约束(number restriction)也不会增加原描述逻辑的计算复杂度。

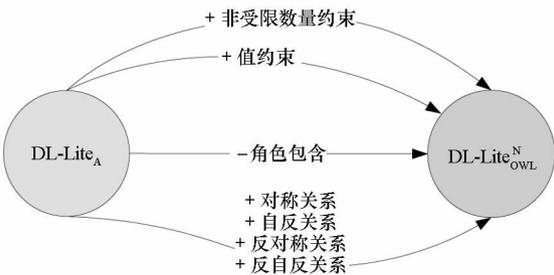


图 1 从 DL-Lite_A 到 DL-Lite^N_{OWL}

Fig. 1 From DL-Lite_A to DL-Lite^N_{OWL}

DL-Lite^N_{OWL} 描述逻辑子集兼顾了 DL-Lite_A 和 OWL 本体语言中的一些重要特性, 同时数据复杂度仍然保持了较低水平。

定义 2 DL-Lite^N_{OWL} 本体可以定义为一个三元组 $O = \langle \Gamma, T, A \rangle$, 其中:

(1) 符号集 $\Gamma = C \cup Op \cup Dp \cup V$, 其中, 概念 C

$= T_C \cup B \cup \neg B \cup \delta(Dp) \cup \exists Op. C \cup \forall Op. C$ (T_C 为顶概念, B 为原子概念, $\delta(Dp)$ 为 Dp 的定义域); 对象属性集合 $Op = C \times C$; 数据类型属性集合 $Dp = C \times V$ 。

(2) TBox T 是一个有限的公理集合, 公理类型包括以下几种:

① 概念包含公理: $C_1 \subseteq C_2$ 。

② 属性约束公理: 包括函数性公理 $funct(Op)$ 和 $funct(Dp)$ 、对称性公理 $sym(Op)$ 、反对称性公理 $asym(Op)$ 、自反公理 $ref(Op)$ 和反自反公理 $irr(Op)$ 。

- ABox A 为一个有限的断言集合。

定义 3 对于给定的 DL-Lite^N_{OWL} 本体 O , O 的解释为 $I = (\Delta^I, \cdot^I)$, 其中:

(1) Δ^I 为解释域, Δ^I 由对象域 Δ'_o 和取值域 Δ'_v 两个互不相交的部分组成, 即 $\Delta^I = \Delta'_o \cup \Delta'_v$ 。

(2) \cdot^I 为解释函数, 该函数为本体 O 中的公理和断言赋予形式化的含义。

为了便于进行查询的处理, 将知识库的 TBox T 划分为仅含有肯定包含公理的 T_D 和含有其他约束公理的 T_{Cst} 两个部分。同时, 为了适应这种划分又采用一种动态 ABox 结构, SCORDA 系统的基本结构如图 2 所示。当进行查询应答时, 首先基于 T_D 抽取与用户查询相关的对象信息动态形成一个 ABox (即动态 ABox), 随后通过本体推理进一步求精, 返回满足全部公理约束的应答。

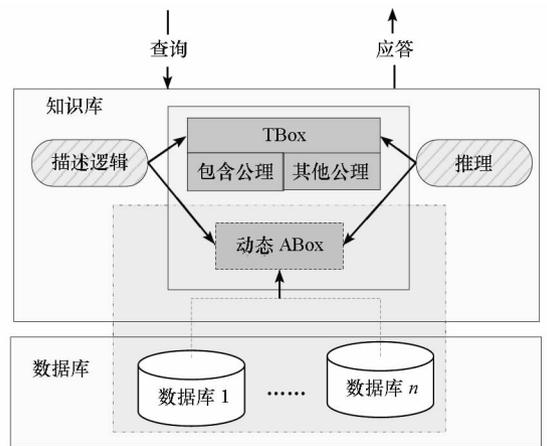


图 2 SCORDA 的知识表示模型

Fig. 2 Knowledge representation model of SCORDA

动态 ABox 仅在查询应答过程中出现, 当查询任务结束, 则 ABox 也随之释放。采用动态 ABox 的 SCORDA 系统呈 3 种状态:

(1) 由 T_D 和 A_\emptyset 构成的初始状态, 记作 K_0 。

(2) 由 T_D 和 A_D 构成的第一阶段查询处理状态, 记作 K_1 。其中 A_D 中的断言为针对 T_D 中的公理获得的结果集。

(3) 由 $T_D \cup T_{Cst}$ 和 A_D 构成的第二阶段查询处

理状态,记作 K_2 。

在 K_0 状态下, DL-Lite_{OWL}^N 知识库仅由 T_D 组成,由于 T_D 中仅包含肯定包含公理,此时知识库 $\langle T_D, A_\phi \rangle$ 中不存在不一致;在 K_1 状态下, A_D 已经非空,此时知识库中 $\langle T_D, A_D \rangle$ 仍然不存在不一致。在 K_2 状态下,知识库 $\langle T_D \cup T_{Cst}, A_D \rangle$ 可能存在不一致,此时需要对 A_D 中的个体进行筛选。

定义4 对于 SCORDA 系统的任意查询 q , 满足 q 的确定应答 $CAns(q, O)$ 是动态 $ABox$ 中的一个常量元组 t , 且 t 中包含 VO 中对象的全部或部分信息, 对于本体 O 的每一个模型 $Mod(O)$, 都有 $t \in q'$ 。

如果 O 的一个解释 I 能够满足 O 中的所有公理和断言, 则 I 为 O 的一个模型。为了保证数据访问的语义正确性, 为本体找到模型是进行数据访问的关键步骤, 只有满足本体模型语义的数据才能够提交给用户, 这也是确保数据语义正确性的必要环节。

2 映射的构建

目前, 关系数据库和本体之间的映射并没有统一的表达形式, 根据具体应用的不同, 映射所含有的语义信息的量级也有所不同。为了克服关系数据库模式和本体间固有的阻抗失配问题, 通常需要使用 GAV、LAV 和 GLAV 等形式的复杂映射表达式^[3,8-9]。

SCORDA 方法首先需要构建关系数据库到本体的 LAV 映射, 然后基于该 LAV 映射自动构造 GAV 映射。其中, LAV 映射的构造过程本质上是对关系数据库语义的挖掘和描述, 使用逻辑公式表达出每个关系模式的语义; GAV 映射的构造则将本体论域上的常量对象和关系数据库中的元组对应起来。而且, 从关系数据库到本体的 LAV 映射可以借助映射发现算法以半自动化的方式生成, 而 GAV 映射的构造过程可以基于 LAV 映射和相应的算法自动完成。这种结构最大的优点在于整合了 LAV 和 GAV 的优势, 每当有新的关系数据源加入或者本体发生演化时, 系统能够很快自适应。

2.1 LAV 映射的发现

LAV 映射本质上就是将关系数据库模式上关系数据库模式的语义准确“投射”到本体之上, 使用本体中的元素将隐含于关系数据库模式中的语义显式表示出来。针对 DL-Lite_{OWL}^N 本体的基本特点, 本文采用一种基于概念连接图的映射发现

方法。

定义5 关系数据库模式 S 到本体 O 的 LAV 映射 M_L 定义为一个形如 $T(X) \rightarrow \Psi(X, Y)$ 的一阶逻辑表达式。其中, $T(X)$ 表示 S 中的一个关系模式; $\Psi(X, Y)$ 表示本体上的一个合取公式, 所有的谓词均来自 O ; 变量集合 X 代表关系模式中的列, 而变量集合 Y 代表 DL-Lite_{OWL}^N 本体论域上的对象。

定义6 如果 DL-Lite_{OWL}^N 本体中的某一元素 $e \in C \cup Op \cup V \cup Dp$ 在公理 $\alpha \in T_D$ 和 $\beta \in T_D$ 中都出现, 则称公理 α 和公理 β 关于 e 结构连接。

定义7 DL-Lite_{OWL}^N 本体 T_D 中的公理构成了一个概念连接图 $G_C = \langle CNodes, Edges \rangle$, 其中图的节点均为本体中的概念, 图的边可以为包含公理或者结构连接关系。

假设有如下的 DL-Lite_{OWL}^N 本体片段, 则相应的概念连接图片段如图3所示, 图中

Conference \subseteq Event $\quad \exists$ OrgnizerOf \subseteq Event
 Person $\subseteq \exists$ OrganizerOf \quad participantOf \subseteq Event
 Person $\subseteq \exists$ participantOf $\quad \rho(\text{hasStartTime}) \subseteq$ Event
 Chair \subseteq Person $\quad \rho(\text{endTime}) \subseteq$ Event
 $\rho(\text{name}) \subseteq$ Person $\quad \rho(\text{eventTitle}) \subseteq$ Event

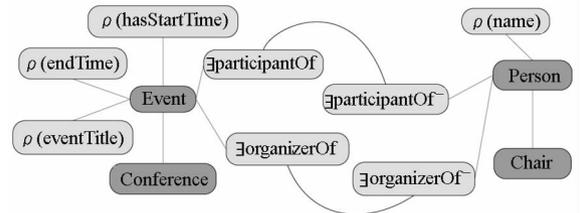


图3 概念连接图片段示例

Fig. 3 Example of concept connection graph

基于概念连接图发现 LAV 映射的基本步骤为:

(1) 建立关系数据库模式中的符号到本体中的符号的对应。假设有前述示例公理集和关系模式 **chair** (name, eventTitle, startingTime, endTime), 则可以添加对应:

DB. **chair**. name \leftrightarrow Onto. Chair. name

DB. **chair**. eventTitle \leftrightarrow Onto. Conference. eventTitle

DB. **chair**. startingTime \leftrightarrow Onto. Conference. hasStartTime

尽管当前模式匹配算法在精度上有了较大提升, 但是仍然无法确保匹配结果的完全正确, 通常需要人工参与对应构建过程。后文中假设已经获得了符合要求的对应。

(2) 标识关系模式对应的概念。概念的标识利用主外键约束和上一步给定的对应集合实现, 例如 **chair** 表反映了多对多关系, 根据用户输入

的对应,可以标识出隐含实体集 *Chair* 和 *Conference*。相关内容可参阅文献[10-11]。

(3) 基于概念连接图进行最小公理集搜索。当关系模式标识出多个概念时,为了形成闭合的语义关系,需要从本体中抽取一个连接这些概念的最小公理集。例如在 DL-Lite_{OWL}^N 中,概念 *Chair* 和 *Conference* 之间的最小公理集为:

$$\text{Chair} \subseteq \text{Person} \quad \exists \text{OrganizerOf} \subseteq \text{Person}$$

$$\exists \text{OrganizerOf}^- \subseteq \text{Event} \quad \text{Conference} \subseteq \text{Event}$$

(4) 将一阶逻辑映射公式编码。编码一阶逻辑形式的 LAV 映射。例如:

$$\mathbf{chair}(name, eventTitle, startingTime, endTime) \rightarrow \text{Chair}(x_1) \wedge$$

$$\text{Person}(x_2) \wedge \text{isa}(x_1, x_2) \wedge \text{name}(x_2, name) \wedge$$

$$\text{Conference}(y_1) \wedge \text{Event}(y_2) \wedge \text{isa}(y_1, y_2) \wedge$$

$$\text{eventTitle}(y_2, eventTitle) \wedge \text{hasStartTime}(y_2, startingTime) \wedge$$

$$\text{hasEndTime}(y_2, endTime) \wedge \text{OrganizerOf}(x_2, y_2)$$

由于存在“SubClassOf”关系的概念之间存在属性的继承关系,因此上述表达式也可以进行语义关系的合并,进一步简化为如下形式:

$$\mathbf{chair}(name, eventTitle, startingTime, endTime) \rightarrow \text{Chair}(x) \wedge$$

$$\text{name}(x, name) \wedge \text{Conference}(y) \wedge$$

$$\text{hasStartTime}(y, startingTime) \wedge$$

$$\text{hasEndTime}(y, endTime) \wedge \text{OrganizerOf}(x, y)$$

2.2 GAV 映射的构造

SCORDA 方法的基本思想就是借助关系数据库和本体间的映射从关系数据库恢复常量对象集,进而利用对象间的语义关系提高查询的语义正确性。从 LAV 映射生成 GAV 映射的常用方法是引入 Skolem 函数,根据键约束生成对象标识符。

(1) 涉及单个概念的 LAV 映射。假设已有如下 LAV 映射(关系表中未参与映射的列记作“_”):

$$\mathbf{person}(name, _, _, email) \rightarrow \text{Person}(x) \wedge$$

$$\text{name}(x, name) \wedge \text{email}(x, email)$$

假设 Skolem 函数 ff , 可以得到如下 GAV 映射:

$$\text{Person}(ff(name), \text{email}(ff(name), email)) \rightarrow$$

$$\mathbf{person}(name, _, _, email)$$

(2) 涉及两个概念及概念间关系的 LAV 映射。以前述 **chair** 表形成的 LAV 映射为例,假设 Skolem 函数 f 和 g , 则可以恢复出如下 GAV 映射:

$$\text{Chair}(f(name)) \rightarrow \mathbf{chair}(name, _, _, _)$$

$$\text{Event}(g(eventTitle), \text{hasStartTime}(g(eventTitle),$$

$$startingTime), \text{hasEndTime}(g(eventTitle), endTime))$$

$$\rightarrow \mathbf{chair}(_, eventTitle, startingTime, endTime)$$

$$\text{OrganizerOf}(f(name), g(eventTitle)) \rightarrow$$

$$\mathbf{chair}(name, eventTitle, _, _)$$

由于 **chair** 表中的外键 $name$ 参考了 **person** 表中的主键 $name$, 为了恢复出 **chair** 对象集的完整信息,还需要对 **chair** 表和 **person** 表进行右外连接:

$$\text{Chair}(f(name), \text{email}(ff(name), email)) \rightarrow$$

$$\Pi_{\text{person.name}, \text{person.email}}(\mathbf{chair}(name, _, _, _)$$

$$\mathbf{person}(name, _, email))$$

(3) 涉及三个以上概念的 LAV 映射的处理与(2)类似,只是需要进行多个对象集的恢复。

由于在 LAV 映射构造过程中已经对数据库语义进行了正确解释,上述映射转换可以自动完成,节省了较多工作量。在具体实现时, GAV 映射右端通常是一个 SQL 语句,前述映射可以写作:

$$\text{Chair}(f(name), \text{email}(ff(name), email)) \rightarrow$$

$$\text{select name, email from person right outer}$$

$$\text{join chair on person.name = chair.name}$$

3 查询处理及实例检验

基于 SCORDA 方法构造的信息系统能够应答合取查询(Conjunctive Query, CQ)、合取查询的并(Union of Conjunctive Query, UCQ)和布尔查询(Boolean Query, BQ)。其中, CQ 是最基本的查询表达形式,它对应关系数据库的 SPJ (Select Project Join) 查询,能够满足常规用户的数据访问需求,其定义如下:

定义 8 DL-Lite_{OWL}^N 本体上的合取查询是形如:

$$Q(x) : - \bigwedge_{0 < i}^n \text{Atom}_i(x, y)$$

的表达式,其中,合取项 Atom_i 可以为 $C(x)$ 、 $Op(x, y)$ 、 $Dp(x, v)$ 等形式的原子, $x, y \in \Delta_0'$ 且 $v \in \Delta_0'$ 。此外,出现在查询头部的变量为 distinguished 变量,在多个原子中都出现的变量为共享变量。

例如查询 $Q(x) : - \text{paper}(x) \wedge \text{hasAuthor}(x, z)$ 的语义就是查找一个属于概念 *paper* 的对象,并且它的 *hasAuthor* 属性为任意值。

UCQ 查询是形如 $Q(x) : - \text{CQ}_1 \vee \text{CQ}_2 \vee \dots$ 的查询, UCQ 查询的应答集等价于各析取项(CQ)应答集的并。布尔查询是形如 $Q : - \text{paper}(x) \wedge \text{hasAuthor}(x, z)$ 的查询,这类查询并不返回对象,仅返回布尔值,其用途为询问知识库中是否存在查询表达式所描述的事实。

考虑到 DL-Lite_{OWL}^N 本体遵循命名唯一性假设,要采用常规 DL 推理机进行实例检验就必须考虑对象 IRI 的动态分配问题。通过 GAV 映射

表达的每一个对象集的 URI 为“本体 URI + 数据库名 + 对象集标识名”,这样即可确保标识符的唯一性。参考文献[12]给出如下定义:

定义9 虚拟对象模式可以定义为一个三元组 $VScheme = \langle Cls, Attri, URI \rangle$, 其中: Cls 为虚拟对象的类名集合, $Attri$ 为虚拟对象的特性集合, URI 为标识虚拟对象的 URI。

定义10 虚拟对象模式的数据库实例定义为 $J = \langle VO, M_C, \mu \rangle$, 其中, VO 为一个有限的虚拟对象集合, 映射 M 将 DB 中的某一元组指派给 VO 中某一对象, 函数 μ 为 VO 的每一个虚拟对象分配一个不同的 URI。

虚拟对象并非 Δ_o^I 上实际存在的对象, 在数据访问过程由关系数据库中的元组动态填充生成。查询处理过程分为3个阶段:

(1) 用户查询的扩展重写阶段。前文已经提到, SCORDA 知识库的 TBox 部分由 T_D 和 T_{Cst} 两部分组成。在查询处理过程中 T_D 中的公理首先参与用户查询 q 的扩展重写环节, 得到重写后的 UCQ 查询 q' 。该过程可参见文献[6]中的 PerfectRef 算法。

(2) 动态 ABox 构造阶段。如图4, 查询 q' 对应的是一个对象集合, 而该对象集合是通过数据库中的数据来构造的。此时, 需要基于查询 q' 和 GAV 映射从关系数据库中获取相关的数据, 为每个对象集分配 IRI, 构成动态 ABox。

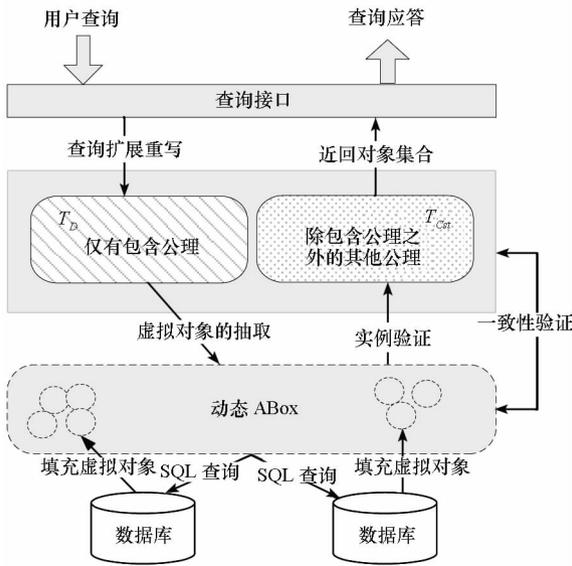


图4 查询处理流程

Fig. 4 The flow of query processing

(3) 实例检测阶段。由于查询的模式重写过程仅仅使用了 TBox 中的部分公理, 因此无法确保获取到的应答集能够满足所有的公理约束, 即无法保证当前知识库一致。实例检验过程的任务是

通过本体 ABox 推理将不符合 T_{Cst} 中公理约束的对象筛选掉。与 OBDA 方法等相比, SCORDA 方法可以采用常规 DL 推理机进行实例检验, 具有更好的可伸缩性。

4 原型系统实现

SCORDA 系统可分为映射管理模块和数据访问模块两部分分别予以实现。其中, SMMS (Semantic Mapping Management System) 映射管理模块如图5所示, 在实现过程中重用了 MAPONTO^[11] 系统中关系数据库模式操作的部分代码, 详细的映射发现算法参见文献[10]。



图5 SMMS 语义映射构造系统

Fig. 5 Semantic mapping management system

在数据访问模块的实现过程中使用 MAPONTO 实验数据集中的 VLDB 数据库和 D2RQ^[13] 示例中的 ISWC 数据库为数据源, 采用 MAPONTO 实验数据集中的 conference. owl 本体作为概念视图 (精简为 DL-Lite_{OWL}^N 本体)。本体操作和推理通过 OWL API 和 Pellet 推理机完成。例如, 假设用户查询为 $Q(x, y) : -hasLocation(x, y)$, 则系统将返回如图6所示的结果, 图中的右半部分界面给出了本体的继承结构, 便于用户构造查询。



图6 数据访问模块

Fig. 6 Data access module

SCORDA 方法利用关系数据库模式的内涵语义进行数据访问,并且通过本体的模型论语义保证数据访问的正确性,因而数据访问的语义正确性是有保证的。限于文章篇幅,本文中仅对 SCORDA 方法的基本框架、查询处理流程和可行性等进行了阐述和验证,关于数据规模和查询响应时间方面的性能比较和论证将在后续文章中给出。

5 结束语

为了促进信息优势向决策优势转换,本文提出了一种面向语义正确性的关系数据库访问方法 SCORDA,给出了完整的解决方案并且验证了其可行性。与已有的 OBDA 等方法相比,该方法对 DBMS 和数据联合工具的依赖度大大下降,并且能够借助常规 DL 推理机完成实例检测推理任务。下一步的工作将就不同数据规模下 SCORDA 方法的稳定性能进行研究,对原型系统做进一步完善。

参考文献 (References)

[1] Chang K C, He B, Li C, et al. Structured databases on the Web: observations and implications [J]. SIGMOD Record. 2004, 33(3): 61 - 70.

[2] 唐富年,姚莉,李金洋. 基于本体的关系数据库访问研究进展[J]. 小型微型计算机系统,2011,32(3):390 - 396.

TANG Funian, YAO Li, LI Jingyang. Ontology-based relational database access: a survey and future direction[J]. Journal of Chinese Computer System, 2011, 32 (3): 390 - 396. (in

Chinese)

[3] Borgida A, Mylopoulos J. Data semantics revisited [C] // Proceedings of the Workshop on Semantic Web and Databases (SWDB) in Conjunction with the International Conference on Very Large Data Bases (VLDB), 2004:9 - 26.

[4] Poggi A, Lembo D, Calvanese D, et al. Linking data to ontologies [J]. Journal on Data Semantics X. 2008: 133 - 173.

[5] Rodriguez-muro M, Calvanese D. Dependencies: making ontology based data access work in practice [C] // Proceedings of the 5th Alberto Mendelzon International Workshop on Foundations of Data Management. Santiago, Chile; 2011.

[6] Calvanese D, de Giacomo G, Lembo D, et al. Tractable reasoning and efficient query answering in description logics: the DL-Lite family [J]. Journal of Automated Reasoning, 2007.

[7] Artale A, Calvanese D, Kontchakov R, et al. The DL-Lite family and relations [J]. Journal of Artificial Intelligence Research, 2009, 36: 1 - 69.

[8] Calvanese D, de Giacomo G, Lembo D, et al. Using OWL in data integration [M]. Semantic Web Information Management—A Model Based Perspective, Springer, 2009.

[9] Poggi A, Rodriguez-muro M, Ruzzi M. Ontology- based database access with DIG-mastro and the OBDA plugin for protégé [C] // Proceeding of the Workshop OWLED. 2008.

[10] Tang F N. Discovering semantic mappings for ontology-based data access [C] // ICDDM 2011, SanYa, China; 2011.

[11] An Y. Discovering and using semantics for database schemas [D]. Toronto: University of Toronto, 2007.

[12] Calvanese D, Lenzerini M, Nardi D. Unifying class-based representation formalisms [J]. Journal of Artificial Intelligence Research, 1999, 11 (2): 199 - 240.

[13] Bizer C, Seaborne A. D2RQ-treating non-RDF databases as virtual RDF graphs [C] // The 3rd International Semantic Web Conference, 2004.