

一种从关系数据库学习 OWL 本体的方法

许卓明,王 琦

(河海大学计算机及信息工程学院,江苏 南京 210098)

摘要 提出了一种从关系数据库学习 OWL 本体的方法.该方法在形式化表示关系数据库模式和 OWL 本体的基础上,遵循从关系数据库模式到 OWL 本体的一组通用映射规则.基于 Java 2 平台的原型工具实现和案例研究证明了该方法的有效性.

关键词 本体学习;关系数据库模式;本体工程;OWL;语义 Web

中图分类号 TP311 **文献标识码** A **文章编号** 1000-1980(2006)02-0208-04

本体^[1]是语义 Web^[2]的关键技术.由于用手工方式开发 Web 本体是一个繁重和棘手的任务,极易导致知识获取的瓶颈,所以本体学习^[3]技术应运而生.关系数据库是基于 Web 的数据密集型应用的主要信息源,数据库模式中隐含着领域知识.因此,从关系数据库学习出的 OWL^①本体更适合于数据密集型 Web 应用的需要.研究从关系数据库学习 OWL 本体的方法,对数据密集型 Web 站点向语义 Web 迁移^[4]、动态 Web 页语义标注^[5]、构建新一代信息管理基础结构^[6]等均具有现实意义.本文介绍一种从关系数据库学习 OWL 本体的方法.

1 关系数据库模式

基表结构设计和完整性约束申明两部分构成了关系数据库模式的主要内容^[7].基于本文研究的问题,这里强调模式中不同类型的基表、基表间的键引用关系以及特殊化关系.定义 1 给出了关系数据库模式.

定义 1 一个关系数据库模式 $S = (L, pkey, unique, notnull, fkey, subof)$ 是六元组,其中:

- a. 名集 $L = E \cup R \cup D$ 是一个有限集,由两两不相交的集合组成,一个实体表(Entity Table)名的集合 E ;一个联系表(Relationship Table)名的集合 R ;一个数据类型(Data type)名的集合 D ,每个数据类型名是 DBMS 预定义的类型名.
- b. $\forall T \in E \cup R, T$ 有一个非空的列的集合 $att(T)$,且每个列 $A \in att(T)$ 有一个相关的预定义数据类型 $type(A) \in D$ 作为其取值范围,其中 $type(*)$ 表示‘*’的预定义数据类型.
- c. $\forall T \in E \cup R, T$ 有且仅有一个主键 $pkey(T)$ (非空的列的集合),要么 $pkey(T) \in att(T)$ (称 $pkey(T)$ 为单主键,此时 T 是实体表),要么 $pkey(T) \subseteq att(T)$ (称 $pkey(T)$ 为复合主键,此时 T 是联系表).
- d. $\forall T \in E \cup R$,每个列 $A \in att(T)$,若 A 取值唯一,则布尔函数 $unique(A) = True$,否则 $unique(A) = False$.
- e. $\forall T \in E \cup R$,每个列 $A \in att(T)$,若 A 取值非空,则布尔函数 $notnull(A) = True$,否则 $notnull(A) = False$.
- f. $\forall T \in E \cup R, T$ 可能有一个或多个引用其他实体表 $G \in E$ 主键的外键 $fkey(T, G)$ (列的集合), $fkey(T, G) \in att(T)$ 且 $value(fkey(T, G)) \subseteq value(pkey(G)) \cup \{null\}$,其中 $value(*)$ 表示‘*’的值域.
- g. 对 $\forall T, G \in E$,若 T 引用 G 主键的外键同时又作为 T 的主键,则称 T 和 G 之间存在子/超关系, T 为 G 的子实体表, G 为 T 的超实体表,此时布尔函数 $subof(T, G) = True$,否则 $subof(T, G) = False$.若干个连续的子/超实体表对构成一个子/超实体表层次.

收稿日期 2005-11-21
基金项目 国家自然科学基金资助项目(60573098)
作者简介 许卓明(1965—),男,江苏苏州人,教授,主要从事数据库、信息检索及语义 Web 技术的研究.
① DEAN M, SCHREIBER G. OWL Web ontology language reference (W3C Recommendation). <http://www.w3.org/TR/owl-ref/>, 2004-02-10/2005-11-20.

2 OWL 本体

本体是一组描述某领域内概念、概念属性以及概念间关系的词汇和公理的集合. W3C 在 2004 年 2 月发布了标准化的 Web 本体语言 OWL,详见 W3C 有关文献. 这里给出 OWL 本体的简化定义(见定义 2).

定义 2 一个 OWL DL 本体 $O = (Cept, Axiom)$ 是二元组,其中:

a. 标识符集 $Cept = CID \cup DPID \cup OPID \cup DTID$ 是一个有限集,由两两不相交的集合组成,一个类(Class)标识符集 CID ;一个数据类型属性(Datatype Property)标识符集 $DPID$;一个对象属性(Object Property)标识符集 $OPID$;一个数据类型(Datatype)标识符集 $DTID$. 每个数据类型标识符是 OWL 本体中使用的预定义的 XML Schema 数据类型标识符^①.

b. 公理集 $Axiom = CAxiom \cup PAxiom$ 是一个有限集,由两两不相交的集合组成,一个类公理(Class Axiom)集 $CAxiom$,包含本体中定义的所有类公理;一个属性公理(Property Axiom)集 $PAxiom$,包含本体中定义的所有属性公理. 表 1 为本文使用到的部分 OWL 构造子的抽象语法.

表 1 OWL(部分)构造子的抽象语法
Table 1 Abstract syntax for OWL constructs (partial)

抽象语法格式	解释
Descriptions(C)	描述(用 C 表示,可以带下标)
A (URI reference)	A 是 URI 引用,用来标识一个命名类
$restriction(R\ cardinality(n))$	属性约束,通过指定对象属性 R 的基数是 n 来定义
$restriction(R\ allValuesFrom(C))$	属性约束,通过指定对象属性 R 全部取值于 C 来定义
$restriction(U\ cardinality(n))$	属性约束,通过指定数据类型属性 U 的基数是 n 来定义
$restriction(U\ maxCardinality(n))$	属性约束,通过指定数据类型属性 U 的最大基数是 n 来定义
$restriction(U\ allValuesFrom(D))$	属性约束,通过指定数据类型属性 U 全部取值于 D 来定义
Data Ranges(D)	数据值域(用 D 表示,可以带下标)
D (URI reference)	D 是 URI 引用,用来标识一个数据值域(预定义的 XML Schema 数据类型)
Object Properties (R)	对象属性(用 R 表示,可以带下标)
R (URI reference)	R 是 URI 引用,用来标识一个对象属性
Datatype Properties (U)	数据类型属性(用 U 表示,可以带下标)
U (URI reference)	U 是 URI 引用,用来标识一个数据类型属性
Property Axioms	属性公理
$DatatypeProperty(U\ domain(C_1)\cdots domain(C_n)\ range(D_1)\cdots range(D_m)\ [Functional])$	数据类型属性 U 的定义域是 $C_1 \cap \cdots \cap C_n$,值域是 $D_1 \cap \cdots \cap D_m$. 如果 U 取值唯一,则 U 是函数属性
$ObjectProperty(R\ domain(C_1)\cdots domain(C_n)\ range(C_1)\cdots range(C_m)\ [inverseOf\ R_0]\ [Functional])$	对象属性 R 的定义域是 $C_1 \cap \cdots \cap C_n$,值域是 $C_1 \cap \cdots \cap C_m$; R 可以是对象属性 R_0 的逆. 如果 R 取值唯一,则 R 是函数属性
Class Axioms	类公理
$Class(A\ partial\ C_1\cdots C_n)$	A 是 $C_1 \cap \cdots \cap C_n$ 的子类
$SubClassOf(C_1\ C_2)$	C_1 是 C_2 的子类

3 从关系数据库学习 OWL 本体的方法

从关系数据库学习 OWL 本体的可行性基于以下事实:运用数据库逆向工程(database reverse engineering)方法可从关系数据库模式提取 ER 模式^[8];ER 模式可语义保持地转换成 OWL 本体^[9]. 因此,可定义一组直接从关系数据库模式到 OWL 本体的映射规则,基于这样的映射规则可开发相应的本体学习器.

3.1 映射规则

为了表示从关系数据库模式到 OWL 本体的映射规则,需要预定义以下辅助函数.

- a. $IS(X)$ 布尔函数. 若 X (表达式)成立,则 $IS(X) = True$; 否则 $IS(X) = False$.
- b. $idMap(ID)$ 将关系数据库模式中表名和属性名映射为 OWL 本体中的同名标识符. 即若 ID 是关系数据库模式中的表名和属性名,则 $idMap(ID) = ID \in CID \cup DPID \cup OPID$.
- c. $dtMap(DT)$ 将关系数据库模式中数据类型名映射为 OWL 本体中相应的数据类型标识符. 即若 DT

① BIRON P V, PERMANENTE K, MALHOTRA A. XML Schema Part 2: Datatypes Second Edition (W3C Recommendation). <http://www.w3.org/TR/xmlschema-2/>, 2004-10-28/2005-11-20.

是关系数据库模式中的数据类型名,则 $dtMap(DT) = DType \in DTID$.

3.1.1 属性公理的生成规则

规则 1 将关系数据库模式中表的非外键列及其相应的预定义数据类型转化为 OWL 本体中以表对应类为定义域的数据类型属性及其相应的预定义 XML Schema 数据类型. 即: $\forall T, G \in E \cup R \wedge \forall A \in att(T) \wedge \neg IS(A = fkey(T, G)) \rightarrow DatatypeProperty(idMap(A) \text{ domain}(idMap(T)) \text{ range}(dtMap(type(A))))$.

规则 2 将关系数据库模式中表 A 引用表 B(实体表)主键的外键转化为 OWL 本体中一对分别以表 A 对应类和表 B 对应类为定义域的互逆的对象属性,其中以表 B 对应类为定义域的对象属性是函数属性. 即: $\forall T, G \in E \cup R \wedge \forall A \in att(T) \wedge IS(A = fkey(T, G)) \rightarrow ObjectProperty(idMap(A) \text{ domain}(idMap(T)) \text{ range}(idMap(G))) ObjectProperty(inv_idMap(A) \text{ domain}(idMap(G)) \text{ range}(idMap(T)) \text{ inverseOf}(idMap(A)) \text{ Functional})$.

规则 3 将关系数据库模式中表的非外键列中取值唯一列转化的 OWL 本体的数据类型属性声明为函数属性. 即: $\forall T, G \in E \cup R \wedge \forall A \in att(T) \wedge \neg IS(A = fkey(T, G)) \wedge unique(A) \rightarrow DatatypeProperty(idMap(A) \text{ Functional})$.

规则 4 将关系数据库模式中表的外键列中取值唯一列转化的 OWL 本体的对象属性声明为函数属性. 即: $\forall T, G \in E \cup R \wedge \forall A \in att(T) \wedge IS(A = fkey(T, G)) \wedge unique(A) \rightarrow ObjectProperty(idMap(A) \text{ Functional})$.

3.1.2 类公理的生成规则

规则 5 将关系数据库模式中表的非外键列转化为 OWL 本体中的一个类公理. 即: $\forall T, G \in E \cup R \wedge \forall A \in att(T) \wedge \neg IS(A = fkey(T, G)) \rightarrow Class(idMap(T) \text{ partial restriction}(idMap(A) \text{ allValueFrom}(dtMap(type(A))))))$.

规则 6 将关系数据库模式中表 A 引用表 B(实体表)主键的外键转化为 OWL 本体中的 2 个类公理,其中以表 B 对应类为定义域的对象属性的基数为 1. 即: $\forall T, G \in E \cup R \wedge \forall A \in att(T) \wedge IS(A = fkey(T, G)) \rightarrow Class(idMap(T) \text{ partial restriction}(idMap(A) \text{ allValueFrom}(idMap(G)))) , Class(idMap(G) \text{ partial restriction}(inv_idMap(A) \text{ allValueFrom}(idMap(T)) \text{ cardinality}(1)))$.

规则 7 将关系数据库模式中表的取值不为空的列转化为 OWL 本体中的一个类公理. 即: $\forall T \in E \cup R \wedge \forall A \in att(T) \wedge notnull(A) \rightarrow Class(idMap(T) \text{ partial restriction}(idMap(A) \text{ cardinality}(1)))$.

规则 8 将关系数据库模式中表的非外键列中取值可以为空的列转化为 OWL 本体中的一个类公理. 即: $\forall T, G \in E \cup R \wedge \forall A \in att(T) \wedge \neg IS(A = fkey(T, G)) \wedge \neg notnull(A) \rightarrow Class(idMap(T) \text{ partial restriction}(idMap(A) \text{ maxCardinality}(1)))$.

规则 9 若关系数据库模式中 2 个实体表之间存在“子超关系”,则将子实体表转化的 OWL 本体中的类声明为子类,超实体表转化的 OWL 本体中的类声明为父类. 即: $\forall T, G \in E \wedge subof(T, G) \rightarrow SubClassOf(idMap(T) idMap(G))$.

3.2 DB2WO 设计思想

关系数据库模式——OWL 本体转换器 DB2WO 的体系结构如图 1 所示. 基于 DB2WO, 本体学习的过程由下面 3 个子过程组成:

a. 通过读取关系数据库数据字典的方法来提取关系数据库的模式信息^[10].

b. 分析提取出的关系数据库的模式信息,利用以上映射规则将关系数据库模式转换成 OWL 本体输出. 实现时将映射规则设计成相应算法,处理步骤不再赘述.

c. 利用现有本体工程工具对生成的 OWL 本体进行精练、评估和验证.

4 实 例

案例研究验证了本文方法的有效性. 这里给出一个实例: 用 SQL Server 2000 设计的关系数据库模式定义 stu_course.sql, 见表 2, 以此为输入, 经过 DB2WO 处理后输出 OWL 本体 stu_course.owl (结果 OWL 本体见图 2).

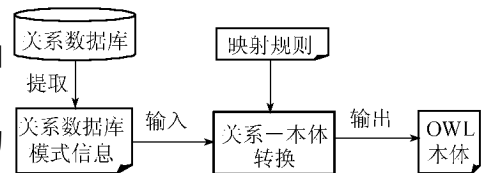


图 1 DB2WO 体系结构

Fig.1 Architecture of DB2WO

表 2 关系数据库模式
Table 2 Relational DB schema

```
CREATE DATABASE stu_course ;
CREATE TABLE Student (
    stuNo char(7) PRIMARY KEY,
    stuName char(12) UNIQUE) ;
CREATE TABLE Teacher (
    tecNo char(7) PRIMARY KEY) ;
CREATE TABLE GStudent (
    gstuNo char(7) PRIMARY KEY,
    instructNo char(7) NOT NULL UNIQUE,
    FOREIGN KEY (gstuNo) REFERENCES Student (stuNo),
    FOREIGN KEY (instructNo) REFERENCES Teacher (tecNo)) ;
CREATE TABLE Course (
    couNo char(6) PRIMARY KEY,
    couName char(20) NOT NULL) ;
CREATE TABLE Enroll (
    estuNo char(7), ecouNo char(6), grade
    int(4), PRIMARY KEY (stuNo, couNo),
    FOREIGN KEY (estuNo) REFERENCES Student (stuNo),
    FOREIGN KEY (ecouNo) REFERENCES Course (couNo)) ;
```



图 2 结果 OWL 本体
Fig.2 Resulting OWL ontology

5 结 语

本文介绍了一种从关系数据库学习 OWL 本体的方法,该方法已在关系数据库模式——OWL 本体转换原型工具 DB2WO 中有效地实现,并在案例研究中得到验证.由于本文方法基于一组‘从关系数据库模式到 OWL 本体’的通用映射规则,DB2WO 基于 Java 2 平台实现,因此具有通用性,工具具有可移植性.

参考文献：

[1] HENDLER J. Ontologies on the Semantic Web[J]. IEEE Intelligent Systems,2002,17(2) :73-74.
[2] BERNERS-LEE T,HENDLER J,LASSILA O. The Semantic Web[J]. Scientific American,2001,284(5) :34-43.
[3] STAAB S,MAEDCHE A. Ontology learning for the Semantic Web[J]. IEEE Intelligent Systems,2001,16(2) :72-79.
[4] STOJANOVIC L,STOJANOVIC N,VOLZ R. Migrating data-intensive Web Sites into the Semantic Web[C]//Proc of the 17th ACM Symposium on Applied Computing. New York :ACM Press,2002 :1100-1107.
[5] HANDSCHUH S,STAAB S,VOLZ R. Annotation for the Deep Web[J]. IEEE Intelligent Systems,2003,18(5) :42-48.
[6] MORENO N,NAVAS I,ALDANA J F. Putting the Semantic Web to work with DB technology[J]. IEEE Data Engineering Bulletin,2003,26(4) :49-54.
[7] 王能斌. 数据库系统教程 :上册[M]. 北京 :电子工业出版社,2002 :22-238.
[8] CHIANG R H L,BARRON T M,STOREY V C. Reverse engineering of relational databases :extraction of an EER model from a relational database[J]. Data & Knowledge Engineering,1994,12(2) :107-142.
[9] XU Zhuo-ming,CAO Xiao ,DONG Yi-sheng,et al. Formal approach and automated tool for translating ER schemata into OWL ontologies [C]//DAI H,SRIKANT R,ZHANG C. Advances in Knowledge Discovery and Data Mining. PAKDD 2004 Proceedings,LNAI Vol. 3056. Berlin :Springer,2004 :464-475.
[10] 许卓明,苏文萍. 关系数据库模式信息的提取[J]. 河海大学学报 :自然科学版,2005,33(2) :202-206.

An approach for learning OWL ontology from relational database

XU Zhuo-ming, WANG Qi

(College of Computer and Information Engineering, Hohai University, Nanjing 210098, China)

Abstract Based on the formal definitions of relational database schema and OWL ontology, the approach follows a set of universal mapping rules from a relational database schema to an OWL ontology. Prototype tool implementation based on Java 2 platform and case study demonstrate the effectiveness of the approach.

Key words ontology learning ; relational database schema ; ontological engineering ; OWL ; Semantic Web