第25卷第1期 华东交通大学学报 Vol. 25 No. 1
2008年2月 Journal of East China Jiaotong University Feb. ,2008

文章编号:1005-0523(2008)01-0061-04

一种本体转换为关系数据库模式的方法

陈 珏,黄兆华

(华东交通大学 信息工程学院,江西 南昌 330013)

摘要:语义网是下一代互联网,本体是语义网描述逻辑的具体实现. 本文提出了一种本体转换为关系数据库模式存储的方法,将本体中的每个类和属性的实例用单独的数据表存储,使用视图来体现对象之间的关系. 本文提出的转换方法的优点是当本体发生修改时,不需要大量修改数据库的结构,适合于中小型本体的转换.

关 键 词:本体;关系数据库;类;属性;视图

中图分类号:TP181 文献标识码:A

语义网(Semantic Web)是现在Web的一个延伸. 当前的Web网络理解为一种语法、句法(Syntactic)网,其功能仅仅是将用户输入的信息按照某种格式显示出来. 语义网(Semantic Web)是按照我们对于信息理解从语法到语义、语用的层次过程,将更复杂的劳动赋予机器完成. 语义Web的基础是对网上内容的描述,它的语义理论基础是描述逻辑. 本体是语义Web描述逻辑的具体实现. 本体是共享概念模型的明确的形式化规范说明[1]. 从本体到关系数据库(relational database,RDB)模式的转换在理论上是可行的. 当前,OWL已成为标准的Web本体语言,因此,本文研究从OWL本体到关系数据库模式的转换方法.

1 OWL本体和关系数据库模式

OWL(Web Ontology Language)是W3C本体论工作小组提出的Web本体论语言规范,有三个子语言:OWL Lite、OWL DL、OWL Full,这三种语言的语义表达能力是递增的. OWL其本质上是一种特殊的RDF,而RDF可以看成是一组3元组,每个3元组由一个主体,一个属性名和一个客体组成. 一些带有属性的类通过OWL概要定义,给出对应于其它表示方法的类型和分类. 所有的类都是owls:Class的子类,通过owls:subClassOf,类之间可以有继承关系. owls:Property用于定义类的属性. 属性是类的资源. 属性通过owl:domain,owl:range等定义语义. owls:subPropertyOf指定属性间的继承关系. OWL对象的这些特点使之在某些方面类似于面向对象技术.

一个关系数据库模式由一组关系模式组成,其中包含数据库的基表结构和完整性约束两部分. 基表结构定义了关系(表)的结构、属性(列)及其数据类型与长度等;完整性约束定义了语义施加在数据上的约束,在此,仅考虑主键和外键约束.

2 从OWL到关系数据库的转换

2.1 OWL中实体的转换

在转换过程中,OWL中的每个类和属性对应于关系型数据库的一个独立的表,类和属性的名字作为表的名称

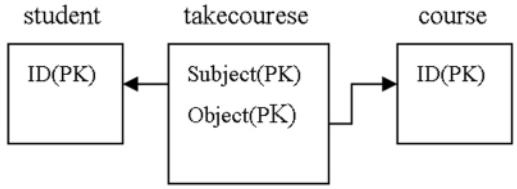
第一步:建立类表. 把OWL对象的所有实体按照其所属的类存储在数据库的对应表中. 表概要地反映了在OWL对象中整个类的层次. 表的每一行对应于类的一个实体,实体的ID,类的名字和Owl对象的所有属性都是数据库表的字段,属性值存储

收稿日期:2007-06-13

作者简介:陈 珏(1980-),女,江西新干人,助教.

在相应的列中. 对于实体记录的不是实体名而是实体的 ID;

第二步: 建立属性表. 属性通过使用属性标的方法记录,记录属性每个实例的主体和客体. 主体的 ID 作为属性表的外键[2]. 如果客体的类型是域或者范围,则客体的 ID 作为属性表的另外一个外键. 如果客体是基本数据类型,其值作为另外一个外键. 对于具有多重值的主体,则在属性表中占据多个行. 如同下图.



2.2 OWL 中关系的转换

OWL 中类和属性的关系很简单,主要用 owls: subClassOf 和 owls: subPropertyOf 表示. 我们用视图来表示这些关系[3]. 通常,视图被用来通过查询语句来创建表和字段的逻辑连接,可以看成是数据库的"虚拟表". 由于在 OWL 中,owls: subClassOf 和 owls: subProperty – Of 是可传递关系,类和属性的视图也应该递归定义. 类的视图可以看成是类直接映射的表和所有子类映射的表的联结. 如在下例中:

< owls: Class owl: ID = "Student" >
< owls: Class owl: ID = "PostGraduate >
< owls: subClassOf owl: Resource = "Student" >
< /owls: Class >
< Student owl: ID = "Jame" / >
< PostGraduate owl: ID = "Jone" >

Student 的视图应该是表 student 和 postgraduate 的联合.

在数据库中存储 OWL 对象时,首先创建对类和属性对应的表,这些表形成了类和属性的初始视图. 一旦创建出初始数据库表和视图,则按照本体对象的层次结构信息调整视图. 调整可以从任何开始,如果这个类有直接的子类,则视图更新为它的相应表和子类的视图的合集. 否则,保持其原来的视图. 对所有类和属性重复此过程,本体数据的层次结构信息将会在数据库中保存并且将来可以从数据库中检索.

3 实例

以下面的 OWL 本体作为范例.

Ontology(studentsTakeCourse Annotation(VersionInfo 1 · 0)
Class(Staff)
DatatypeProperty(staffNo domain (Staff) range (xsd: short) Functional)
DatatypeProperty(staffName domain(Staff) range(xsd: string))
DatatypeProperty(staffAge domain(Staff) range(xsd: integer))
DatatypeProperty(jobTitle domain(Staff) range(xsd: string))
DatatypeProperty(emailAddr domain(Staff) range(xsd: string))
Class(AcademicStaff)
subClassOf(AcademicStaff Staff)
DatatypeProperty(specialty domain(AcademicStaff) range(xsd: string))
Class(Student)
DatatypeProperty (studentNo domain (Student) range (xsd: short) Functional)
DatatypeProperty(studentName domain (Student) range (xsd: string))
DatatypeProperty(major domain(Student) range(xsd: string))
DatatypeProperty(enrollmentDate domain(Student) range(xsd: date))
Class(Course)
DatatypeProperty(courseNo domain(Course) range(xsd: short) Functional)
DatatypeProperty (courseName domain (Course) range (xsd: string))
DatatypeProperty(creditHour domain(Course) range(xsd: integer))
Class(CourseLearning)
ObjectProperty (takenBy domain (CourseLearning) range (Student))
ObjectProperty(inv – takenBy domain(Student)
range(CourseLearning) inverseOf(takenBy))
ObjectProperty (takesCourse domain (CourseLearning) range (Course))
ObjectProperty (inv – takesCoursedomain (Course) range (CourseLearning) inverseOf(takesCourse))
Class(CourseTeaching)
ObjectProperty(taughtBy domain(CourseTeaching) range(AcademicStaff))
ObjectProperty(inv – taughtBy
domain(AcademicStaff)
range(CourseTeaching) inverseOf(taughtBy))
ObjectProperty (teacherOf domain (CourseTeaching) range (Course))
ObjectProperty(inv – teacherOf
domain(Course) range(CourseTeaching) inverseOf(teacherOf))

经过转换处理后,对应的关系数据库模式中部分数据表如下.

(1) Staff 表

实体 ID	Text
StaffNO	smallint
StaffName	int
Staffage	text
JobTittle	text
emailaddr	text

(2) Academicstaff 表

实体 ID	Text
specially	text

(3) student 表

实体 ID	Text
studentNo	Smallint not null
studentName	text
major	text
erollmentDate	date

(4) course 表

实体 ID	Text
courseNo	Smallint not null
courseName	text
creditHour	int

4　结束语

以往,在进行本体向关系数据库转换的过程中,人们常采用的方法有两种,其一是如文献 [3] 所述将所有的实体保存在同一张数据表中,虽然这种方法简单易实现,但是也存在一些缺点. 首先,由于对每一个在 OWL 中描述的属性,都在表中有一个相应得字段. 对于大规模的 OWL 对象,由于有大量的属性,关系型数据库有可能不能存储所有的字段; 其次,OWL 对象的每个实体只有几个属性,但是要占用的数据库表却是一样,只是把没有用的属性设置成空,这将造成空间浪费; 第三,对表的每个字段,只能有一个值. 但是在 OWL 对象中,一个类的一些属性可能有多个值; 第四,OWL 对象中类和属性的层次结构不能在这种方法中描述. 类和属性之间的关系将全部丢失,缺乏可维护. 如果需要添加有新属性的新类或删除某些本体时,则因为需要在表中加入新的字段或删除字段,从而需要修改数据库结构.

其二是同样用一个表存储所有数据. 在这种方法中,表只有 3 个字段: 主体字段存储实体名称,谓词字段存储属性名称,客体字段存储属性的值. 这种方法是关系型数据库存储数据的通用方法. Jena 中使用的就是这种方法,此方法数据库结构简单,易于 OWL 三元组的分析和存储在添加或删除类时,数据库结构不用改变,但是此方法也无法存储类和属性的层次关系,而且,要对数据进行查询,整个表都需要载入.

而文献 [4] 中本体到关系数据库模式的转换其转换规则是基于它们之间的概念对应,其对应关系如表 1,并根据此对应关系分别构造出实体表和联系表. 此方法的不足之处在于需要建立大量的索引,每个本体类转换成一个 InnoDB 类型的表时,主/外键列上都要添加索引定义.

表 1　本体与关系数据库模式之间元素的对应关系

本体中的元素	关系数据库模式中的元素
类	表
以一个类为定义域的数据类型属性及其 XML 模式数据类型	表的非外键列及其相应的预定义数据类型
以一个类为定义域的一个或多个数据类型属性如果是函数属性	其中一个列声明为定义域类所对应的表(实体表) 的主键,其他列声明为唯一列
类 1 与类 2 之间的一对互逆的对象属性,类 1 与类 3 之间的一对互逆的对象属性,且类 2 和类 3 中都有函数属性	类 1 对应的表为联系表,其中添加一个引用类 2 对应的表(实体表) 主键的外键以及一个引用类 3 对应的表(实体表) 主键的外键,并将这两个外键组成这个联系表的复合主键
类 1 与类 2 间的子类 - 超类关系,即类 1 是类 2 的子类	在类 1 对应的表中添加一个主键列,且这个列同时成为引用类 2 对应的表的主键的一个外键

文献 [2] 提出了一种基于关系数据库本体构建方案,其包括关系数据库资源提取、本体概念生成、概念之间关系标注和与已有本体进行集成四个过程. 然后对概念生成过程进行研究,提出先从数据库资源生成中间实体,再将中间实体映射成本体概念的新构想.

本文所提出的是一个与文献 [2] 相反的一个问题,受此文献中所采用的中间实体的启示,本文所采用的方法结合了水平表、垂直表等传统的方法,同时也考虑到了本体中的属性,分别建立了类表和属性表,兼顾了本体信息的存储以及本体中类和属性的关系,通过建立类表和属性表使得本体的修改工作

变的简单易行,不需要大量修改数据库结构,同时利用视图表现了对象之间的关系. 这种方法比较适合中小型的本体,查询速度较快.

当然,本方法也存在不足之处,对于大规模的本体对象,这种方法形成的本体数目会超出数据库允许的范围. 如何有效存储大规模本体的数据和关系仍然是将来探索的一个重要问题.

参考文献:

[1] 宋 炜,张 铭. 语义网简明教程 [M]. 北京: 高等教育出版社,2004.

[2] 韩 石. 基于关系数据库的本体构建方法的研究 [D]. 哈尔滨: 哈尔滨工业大学,2006.

[3] Mehul Bhatt, Andrew Flahive, Carlo Wouters, Wenny Rahayu, David Taniar. A Distributed Framework for Materialized Ontology View Extraction [J]. Algorithmica, 2006, 45(7): 457 – 481.

[4] 许卓明,黄永菁. 从 OWL 本体到关系数据库模式的转换 [J]. 河海大学学报(自然科学版),2006,34(1):95 – 99.

An Approach to Convert Ontology into Relational Database Schema

CHEN Li, HUANG Zhao – hua

(School of Information Engineering, East China Jiaotong University, Nanchang 330013, China)

Abstract: Semantic Web is the next – generation Internet, the Semantic Web Ontology is the realization of Description Logic. In the paper, we present an approach to convert ontology into a relational database schema which stores the instances of each classes and properties in separate tables and uses some views to reflect the relationship between the OWL objects. The proposed converting method has the advantage that it does not require extensive modification of the database structure when changes occur. It is suitable for small and medium – sized ontology change.

Key words: ontology; relational database; class; property; view

(责任编辑: 王建华)