

OWL 本体关系数据库存储模式设计

陶 皖,姚红燕

(安徽工程科技学院 计算机科学与工程系,安徽 芜湖 241000)

摘 要 本体合理有效的存储是保证本体共享利用的前提,现有的关系数据库本体存储模式在存储 OWL 本体后会出现查询推理效率低、维护困难等问题。文中在分析现有模式的基础上,针对 OWL 本体类和属性的特点,通过单独设立类关系表、添加关系约束表 T-restrict 等方法改进已有模式,使其能比较方便地实现 OWL 本体中类、属性及复杂关系的信息存储,学校本体的存储实例表明存储实现简单,查询实验表明查询检索效率有一定提高,从而说明本体存储模式的设计是可行的。

关键词 OWL 本体;关系数据库;存储模式

中图分类号: TP311

文献标识码: A

文章编号: 1673-629X(2007)02-0111-04

Design of OWL Ontology Storage Schema in Relational Database

TAO Wan, YAO Hong-yan

(Dept. of Computer Sci. & Eng., Anhui University of Technology and Sci., Wuhu 241000, China)

Abstract The rational storage of ontology is the premise of the shareable use of ontology. Under current ontology storage schema, some problems such as difficult maintenance, low query and inference speed may appear after OWL ontology storing in relational database. Firstly, current ontology storage methods are analyzed. Secondly, ontology storage schema is designed according to OWL ontology characteristic. The existing design is improved by setting up separate class table, appending relational restrict table T-restrict. As a result, the storage of OWL classes, properties and the complex relations among them is easily realized. Lastly, an example of school OWL ontology storage has indicated that the storage method is simple. The query test has proved that the retrieval efficiency is increased. All these explain that the design is feasible.

Key words OWL ontology; relational database; storage schema

0 引言

OWL 本体^[1]由于 W3C^[2]的推荐已成为语义 Web 研究者的第一选择,这在很大程度上有利于语义 Web 研究的规范。本体是对共享概念化的一种显式的形式化规范说明,典型的本体一般由一组概念、描述概念特征的属性、概念之间的关系、概念和属性之间的约束来表示静态的领域知识,因此一个本体可认为是一种描述某个领域的概念模式,这在本质上决定了本体可以借助较成熟的关系数据库技术来完成本体存储、查询等工作。文中即讨论基于关系数据库的本体存储模式的现状,并在分析现有的存储模式后,以学校 OWL 本体为实例,设计和说明在关系数据库中的存储模式。

1 本体的关系数据库存储模式

本体可以在平面文件(flat file)、数据库等中来存储。这里重点讨论本体的关系数据库存储问题,由于本体模型和关系模型的差异,可以采用多种存储模式在关系数据库中存储本体^[3,4]:

1) 垂直模式。该模式包含一张三元组表,表中的每一条记录对应于一个 RDF 的三元组,即本体中的所有信息都要使用 RDF 的三元组(subject - predicate - object)来表示(如图 1 所示的一种方式,subject 表示某个属性的资源,属性的名称用 predicate 表示,若是数值属性,object 就是属性的值,若是对象属性,object 就是目标资源。在此种表示中,资源的不同属性通过 subject 以相同的资源 URI 联系起来)。这种模式设计简单,并且模式稳定,本体修改时只需修改表中的相应的元组,并且易于通用(本体模型均可以转换为 RDF 模型来表示),但该模式可读性差,本体查询时需检索整个数据库,尤其是在进行表连接查询时效率很低。

收稿日期: 2006-04-27

基金项目: 安徽省高校省级自然科学基金项目(2005KJ065);安徽省高校青年教师科研基金资助项目(2005jqt069)

作者简介: 陶 皖(1972-),女,安徽芜湖人,讲师,硕士,主要从事语义 Web、本体工程研究。

Triples		
Subject	Predicate	Object
(resource URI)	(property name)	(property value)

图 1 垂直模式

2) 分解模式。该模式包含多张表(如图 2 所示),基本思想是对数据库进行模式分解。根据分解对象的不同,分解模式可以有两种:一是基于类的分解模式,为本体的每个类都创建一张单独的表,表名为类名,表的列为类的属性,这种表结构清晰,但很难适应本体动态变化的情况,当本体中类或属性变化时,表结构也要随着变化。二是基于属性的分解模式,即为本体的每个属性创建一张单独的表,表名为属性名,每个表都包含两个列,分别代表 RDF 三元组中的 Subject 和 Object,但该方法对类的隐含实例的查询代价很大。

Property ₁		Class ₁	
Subject	Object	Subject	Object
(resource URI)	(property value)	(resource URI)	(classID)

...

Property _n		Class _n	
Subject	Object	Subject	Object
(resource URI)	(property value)	(resource URI)	(classID)

图 2 分解模式

3) 混合模式。将上述的两种模式进行混合使用(如图 3 所示),为每个不同的对象属性值类型建立三元表(相同类型的放在同一张表中,以 propertyID 或 classID 区别),为类的实例建立二元表。

Properties with range resource			Class Instances	
Subject	Predicate	Object	Subject	Object
(resource URI)	(property name)	(property value)	(resource URI)	(classID)

...

Properties with range integer		
Subject	Predicate	Object
(resource URI)	(property name)	(property value)

图 3 混合模式

通过对本体存储模式分析,可以发现在关系数据库存储中本体模式的设计应满足:

- ① 具有尽可能高的规范化程度(例如满足 3NF 或 BCNF 等);
- ② 易于理解,为了本体应用的方便(如:用于查询)本体设计模式应直观,否则会给设计和带来困难(如:查询语句设计繁琐、效率低);
- ③ 结构稳定,本体是不断进化的,若存储模式随本体变化而变化,数据库系统维护代价太大^[3]。

现在有许多系统扩展了关系数据库的功能,使其能支持本体的存储(如:Jena^[4,5]支持垂直模式,Sesame^[4,6]支持垂直和分解模式,DLDB^[4,7]支持分解模式,Rstar^[4,8]支持垂直模式,RDFSuite^[4,9]支持分解和混合模式,3Store^[4,10]支持垂直模式等)这些系统大多能以一种或一种以上的存储模式支持本体的存储,但这些存储并没有考虑本体的存储模式的适用性,尤其对于大型本体而言在存储的空间效率、查询推理时间、维护代价等方面均存在缺陷,会影响本体的使用。因此在本体的存储上应结合本体自身的特点,进行具体的、更细致的模式设计后再结合关系数据库使用。

2 OWL 本体的存储模式设计

针对第 1 小节的分析,以一个学校的 OWL 本体(如图 4 所示)为实例来分析 OWL 本体的存储问题。

与早期本体描述语言相比,OWL 具有更强的描述能力,它引入了更多的本体描述词汇,例如:Class, property, individual, subClassOf, subPropertyOf, equivalentClass, equivalentProperty, differentFrom 等。这些词汇扩展了语言的查询及推理能力,但在存储时必须合理保存本体类、属性及它们之间的关系(如:本例中的 subClassOf 等),同时要注意关系表的维护、查询连接效率及存储空间等问题,因此在设计存储模式时,应在存储空间、访问时间、推理查询时间等方面取得平衡。

文献[3]中作者对本体存储模式做了较详细的设计,文中的设计即受其启发,但文献[3]的设计存在存储空间占用多以及不能明确描述属性的范围约束(如本例中的 Teacher 类 teacherAge 在 20 到 65 之间;SchoolClass 类 studentNumber 在 1 到 50 之间)等问题,现设计如下:

a. OWL 本体中资源以 URI 来唯一标识,并区分为类(Class)、属性(Property)和实例(Individual),设计一张表 T-resource 将资源 URI、资源名称记录下来,并添加字段 type 记录资源的三种不同类型,以方便对本体信息的索引,从而提高查询效率^[3]。

b. OWL 本体中常使用 subClassOf, subPropertyOf, equivalentClass, equivalentProperty 等来描述类或属性间的关系(层次或等价等关系),为提高查询推理效率,将经常出现的关系组织成单独的表^[3](本例中设计了 T-subClass 表)。

c. 设计一张表 T-domain&range 记录属性的 domain 和 range 信息,设计一张表 T-specProperty 记录其他属性信息(如:InverseFunctionalProperty, FunctionalProperty, TransitiveProperty, SymmetricProperty 等)。

```

<owl :Class rdf :ID=" Student "/>
<owl :Class rdf :ID=" SchoolClass "/>
<owl :Class rdf :ID=" Teacher "/>
<owl :ObjectProperty rdf :ID=" composed_ by ">
  <rdfs :domain rdf :resource=" # SchoolClass ">
  <rdfs :range rdf :resource=" # Student ">
</owl :ObjectProperty>
<owl :Class rdf :ID=" Graduate ">
  <rdfs :subClassOf rdf :resource=" # Student ">
</owl :Class>
<owl :Class rdf :ID=" UnderGraduate ">
  <rdfs :subClassOf rdf :resource=" # Student ">
</owl :Class>
<owl :DataProperty rdf :ID=" studentNumner ">
  <rdfs :domain rdf :resource=" # SchoolClass ">
  <rdfs :range rdf :resource=" &xsd :positiveInteger ">
<owl :minCardinality rdf :datatype=" &xsd :positiveInteger ">1
  </owl :minCardinality>
  <owl :maxCardinality> 50 </owl :maxCardinality>
</owl :DataProperty>
<owl :ObjectProperty rdf :ID=" has_ teacher ">
  <rdfs :domain rdf :resource=" # SchoolClass ">
  <rdfs :range rdf :resource=" # Teacher ">
</owl :ObjectProperty>
<owl :DataProperty rdf :ID=" teacherAge ">
  <rdfs :domain rdf :resource=" # Teacher ">
  <rdfs :range rdf :resource=" &xsd :integer 20...65 ">
</owl :DataProperty>
<Graduate rdf :ID=" Liming ">
</Graduate>
<Teacher rdf :ID=" Wangping ">
  <teacherAge> 30 </teacherAge>
</Teacher>
<SchoolClass rdf :ID=" Computer041 ">
  <stdentNumber> 40 </studentNumber>
  <composed_ by> Liming </composed_ by>
  <has_ teacher> Wangping </has_ teacher>
</SchoolClass>
  
```

图 4 学校 OWL 本体(片断)

d. 设计一张表 T - restriction 记录对属性的各种约束(如 :allValuesFrom ,someValuesFrom ,minCardinality ,maxCardinality 和 cardinality 等),为说明属性值的范围在其中特别设置了 Value 字段,它与 restriction 字段配合即可比较清楚地表明属性的取值情况。

e. 将不经常使用的关系(sameIndividualAs ,differentFrom ,alldifferent ,intersectionOf 等)组成一张表 T - specRelation(本例中未使用),以保证关系的完整,又提高存储的效率,并使得以后的维护方便(当添加新的关

系时,只需添加元组信息,不需要改动表的模式)^{3]}。

根据以上设计,将图 4 所示的 OWL 本体片断做如图 5 所示的存储。

T - resource

Resource URI	Resource name	Resource type
XX/Student	Student	class
XX/SchoolClass	SchoolClass	class
XX/Teacher	Teacher	class
XX/Graduate	Graduate	class
XX/UnderGraduate	UnderGraduate	class
XX/composed_ by	composed_ by	property
XX/studentNumber	studentNumber	property
XX/has_ teacher	hsa_ teacher	property
XX/teacherAge	teacherAge	property
XX/Liming	Liming	Individual
XX/Wangping	Wangping	Individual
XX/Computer041	Computer041	Individual

注 :XX 表示一个 URI 网,如 :http //auts. edu. cn/computer/schoolowl/。

T - restriction

Class	Property	Restriction	Value
School	studentNumber	minCardinality	1
School	studentNumber	maxCardinality	50
Teacher	teacherAge	minvalue	20
Teacher	teacherAge	maxvalue	65

T - domain&range

Property	Domain	Range
composed_ by	SchoolClass	Student
studentNumber	SchoolClass	Integer
has_ teacher	SchoolClass	Teacher
teacherAge	Teacher	Integer

T - subClass

SubClass	SuperClass
Graduate	Student
UnderGraduate	Student

T - individual

Individual	Class
Liming	Graduate
Wangping	SchoolClass

T - invalue

Individual	Property	Value
Wangping	teacherAge	30
Computer041	studentNumber	40
Computer041	composed_ by	Liming
Computer041	has_ teacher	Wangping

图 5 学校 OWL 本体的存储模式

图 5 中的存储表示说明设计易于实现,针对 5 种常用查询要求^[3 4] { 查找某个类的直接父类,查找某个类的所有直接子类,查找某个类的所有直接属性,查找某个类的所有实例(按条件)查找本体中是否包含某个概念 }设计 SQL 语句如下 :

```

SELECT SuperClass FROM T - subClass WHERE SubClass = YY ;
SELECT SubClass FROM T - subClass WHERE superClass = YY ;
SELECT Property FROM T - domain&range WHERE domain = YY ;
SELECT Individual FROM T - individual WHERE Class = YY ;
  
```

SELECT Resourcenname FROM T - resource WHERE YY.

通过学校本体实例的查询实验表明效率较高(与垂直模式相比提高了近 40%) ,说明本体存储的设计是可行的。

3 结束语

本体的合理存储是保证本体共享利用的前提 , OWL 本体扩展了类和属性间的关系 ,能提高查询和推理的能力 ,但丰富的信息如何有效地存储是必须研究的问题 ,文中对 OWL 本体存储模式做了一定的分析和研究 ,但在结合信息查询推理以进一步推进本体存储优化上还存在许多问题 ,这将是以后的研究重点 ,此外实验所采用的数据集还较小 ,引入大数量集后可能出现的问题也将是研究的目标。

参考文献 :

[1] Mcguinness D L ,Harmelen F V. OWL Web Language Overview(W3C Recommendation) [EB/OL] . 2004 - 02 - 10. <http://www.w3.org/TR/2004/REC-owl-features-20040210/>.

[2] W3C. The Semantic Web Home Page[EB/OL] 2006 - 04. <http://www.w3.org/2001/sw/>.

[3] 李 曼. 基于关系数据库的大规模本体的存储模式研究 [J] . 华中科技大学学报 : 自然科学版 , 2005 , 33(12) : 217 - 220.

[4] Theoharis Y. Bechmarking Database Representations of RDF/ S Stores[C] //In Proc. of ISWC2005 ,the 4th International Semantic Web Conference. Galway ,Ireland [s. n.] 2005.

[5] Wilkinson K. Raynolds Efficient RDF Storage and Retrieval in Jena[C] //In Proc. of SWDB2003. Berlin ,Germany [s. n.] , 2003.

[6] Alexaki S. Sesame :A generic Architecture for Storing and Querying RDF and RDF Schema[C] //In Proc. of the ISWC2002 ,the 1st International Semantic Web Conference. Sardinia ,Italia [s. n.] 2002.

[7] Pan Z Heflin J. DLDB :Extending Relational Database to Support Semantic Web Queries[C] //In Proc. of the PSSS2003 , the 1st International Workshop on Practical and Scalable Semantic Web Systems. Sanibel Island ,USA [s. n.] 2003.

[8] Ma L. Rstar :An RDF Storage and Query System for Enterprise Resource Management[C] //In Proc. of the ACM CIKM 2004 ,the 13th Conference on Information and Knowledge Management. Washington D. C. ,USA [s. n.] 2004.

[9] Agrawal S. On Storing Voluminous RDF Descriptions :The case of Web Portal Catalog[C] //In Proc. of WebDB2001 ,the 4th International Workshop on the Web and Databases in conjunction with ACM SIGMOD '01 Conference. Santa Barbara , USA [s. n.] 2001.

[10] Harris S ,Gibbins N. 3Store :Efficient Bulk RDF Storage[C] // In Proc. of the PSSS2003 ,the 1st International Workshop on Practical and Scalable Semantic Web Systems. Sanibel Island , USA [s. n.] 2003.

(上接第 110 页)

个奇异点都为虚假奇异点。

b. 用边长为 $4 * b$ 的矩形封闭边框重新计算每个奇异点的 Poincare Index ,如果得到的类型与原来的不同 ,则为虚假奇异点。

c. 如果奇异点离背景区域或图像边界很近 ,则为虚假奇异点。

3 实验结果及结论

使用几幅指纹图像进行实验 ,实验结果如图 3 所示。



图 3 奇异点提取结果(圆型为核心点 ,矩形为三角点)

根据实验 ,对于不同类型的指纹 ,用该种计算方法可以准确地得到奇异点位置 ,为以后的分类和匹配打下了良好的基础。

参考文献 :

[1] Bazen A M ,Gerez S H. Systematic methods for the computation of the directional fields and singular points of fingerprints [J] . IEEE Trans on pattern Analysis and Machine Intelligence 2002 ,24(7) : 905 - 918.

[2] 沈 伟 ,陈 霞 ,SHEN Jun. 指纹图像奇异点提取的一种鲁棒方法 [J] . 计算机工程 2003 ,29(2) : 16 - 18.

[3] 谭台哲 ,宁新宝 ,尹义龙 ,等. 一种指纹图像奇异点检测的方法 [J] . 软件学报 2003 ,14(6) : 1082 - 1088.

[4] 杨海军 ,梁德群 ,田 原. 基于方向场特征的指纹图像奇异点检测 [J] . 自动化学报 2001 ,27(2) : 273 - 275.

[5] 黄 韧 ,陆亨立 ,张维新. 基于方向角度变化的指纹奇异点提取的快速算法 [J] . 华东师范大学学报 :自然科学版 , 2002(1) : 52 - 55.