第31卷 第3期
2016年6月

北京信息科技大学学报
Journal of Beijing Information Science & Technology University

Vol. 31 No. 3
Jun. 2016

文章编号:1674-6864(2016)02-0059-05 DOI: 10.16508/j.cnki.11-5866/n.2016.03.013

本体存储方法研究

张 慧,侯 霞,李 宁

(1.北京信息科技大学 计算机学院,北京 100101;
2.北京信息科技大学 网络文化与数字传播北京市重点实验室,北京 100101)

摘 要: 通过对现有本体存储方法进行分析,提出了一种利用图数据库来存储本体的方法。图数据库具有天然的图式结构,与本体逻辑结构复杂的图式结构相符,可以在一定程度上解决本体与本体存储介质逻辑结构不匹配的问题。在采用图数据库存储本体的理论基础之上,开发了基于OWL本体和Neo4j的原型系统,可以实现本体到图数据库的完整存储,进而说明本方法的可用性和有效性。

关 键 词: 本体;图式结构;图数据库;本体存储

中图分类号: TP 302 **文献标志码:** A

A research on ontology storage method

ZHANG Hui, HOU Xia, LI Ning

(1. School of Computer, Beijing Information Science and Technology University, Beijing 100101, China; 2. Beijing Key Laboratory of Internet Culture and Digital Dissemination Research, Beijing Information Science and Technology University, Beijing 100101, China)

Abstract: An ontology storage method is put forward based on graph database by analysingexisting ontology storage methods. Graph database has a natural graph structure, which is consistent with complex schematic structure of logic structure of ontology. This method could solve the problem of mismatching of ontology and its storage medium structure. Related experiments based on OWL ontology and Neo4j show the effectiveness of the proposed method and availability.

Key words: ontology; graph structure; graph database; ontology storage

0 引言

语义Web中含有丰富的语义信息,可以被机器理解和有效处理,因此能大幅度提高网络服务效率。本体[1]作为语义Web的关键部分,其存储问题一直以来就受到很多学者和研究者的广泛关注。

本体最早是一个哲学概念,认为是客观存在的系统的说明。如今在人工智能领域,很多学者对本体进行了不同的定义。Neches等[2]将本体定义为"给出构成相关领域词汇的基本术语和关系,并利用这些术语和关系构成的规定这些词汇的外延规则"。Gruber[3]定义本体为"本体是对概念化体系的规范说明"。之后Borst在Gruber给出本体定义的基础之上,重新定义了本体为"共享概念模型的形式化规范说明"[4]。目前用得最多的是由Studer等[5]在深入分析总结前人研究成果基础上得出的本体定义,即本体是共享概念化明确的形式化规范说明。此定义形象地揭示了本体具有概念化、明确性、形式化和共享的4层含义。随着语义Web的快速发展和广泛应用,选择一种合适的本体存储方法显得尤为重要。有效的本体存储方式不仅决定了系统的运用效率,同时也决定了其整体性能。

本文在分析传统本体存储方法的基础之上,提出了采用图式方法对本体进行存储的理论,以避免

收稿日期: 2015-11-10
基金项目: 北京市属高等学校高层次人才引进与培养计划项目(CIT&TCD201504056,CIT&TCD201304115);北京市属高等学校创新团队建设与教师职业发展计划项目(IDHT20130519)
作者简介: 张 慧,女,硕士研究生;通讯作者:侯 霞,女,博士,副教授。

逻辑结构复杂的本体存储在非图式结构存储介质中造成的语义数据不完整或产生冗余数据问题的发生。通过相关的实验验证了该理论的可行性和有效性。

1 相关研究

目前关于本体的存储问题,国内外的很多学者和专家已经进行了相关的研究并取得了一定的成就。主要包括文本式存储、内存式存储、关系数据库式存储以及一些其他方式的存储方法。

1.1 文本式存储

文本式存储是一种原始的存储方法,即采用纯文本的形式对本体进行存储。该存储方法形式简单。目前常用 XML 形式的文件来存储本体,例如常见的 OWL 本体 wine. owl 和 family. owl,以及文献[5]中提及的 Topic Map 本体都是采用这种方式存储的。对于规模较小的本体可以采用文本式方法进行存储,但是当本体规模较大时,往往会使存储信息产生冗余,从而会大大降低系统的运行效率。

1.2 内存式存储

本体的内存式存储是将本体以一定的形式存储在主存中,此时关于本体的操作都可以在主存中进行。此种存储方式的优点是系统运行速度快、效率高。但是由于主存的容量受到限制,不适用于规模较大的本体。这种方式实际是一种通过牺牲容量来得到效率的方法。例如 OWLim[7] 和 OWLJessKB[8] 就是采用这种方式进行本体存储的本体管理系统。

1.3 关系数据库式存储

关系数据库式存储本体的原理是将本体映射为一张或多张二维表,根据映射方式的不同主要分为水平式存储、垂直式存储、分解式存储和混合式存储等模式。关系数据库存储本体的思路基本是利用三元组及其拆分的思想。由于关系型数据库具有成熟的存储机制和对数据的管理能力,因此很多学者都试图将本体存储到关系数据库中。然而关系数据库具有二维表固定结构的特点,往往导致其在存储本体时容易丢失语义信息,或是由于表中数据出现空值,造成存储空间的浪费。更重要的是表连接操作效率低,增加了查询负担。例如文献[9-10]都是基于关系型数据库通过采用不同的方法对本体进行存储,虽然都在不同程度上提高了存储效率,但是本体与关系型数据库结构的不匹配问题仍旧存在。

1.4 其他方式存储

目前对于本体的图式存储也有一些研究,但是没有针对本体进行完整的探讨。文献[11]提出采用属性视图对关系数据库中的数据进行映射,在一定程度上可以避免结构的冗余,但是增加了本体存入关系数据库中的步骤。文献[12]提出利用图数据库的图式模型对海量的 RDF 数据进行分布式存储,但文章只是对一般的 RDF 数据进行了实现,并没有对本体的公理等内容进行存储映射说明。文献[13]采用了解析图的方式对本体中的类等进行了映射,但并没有将公理等关系考虑在内,并且对于大规模本体不适用。

基于上述情况,本文提出了一种基于图数据库的完整本体存储方法,旨在利用图数据库和本体都具有天然图式结构以及图数据库可以存储上亿节点的特点,实现本体到图数据库的具体映射,进而在最大程度上保证本体语义信息的完整性以及避免冗余信息的产生。本文通过相应的实验,验证了方法的可行性和有效性。

2 映射规则

2.1 相关介绍

图数据库具有天然的图结构,在近几年凭借其在高度关联的数据中复杂而动态的联系获得了相应的优势。它是在图论的基础上发展起来的一种新事物。对于存入其中的数据操作,实质就是对图的遍历与搜索。在处理关联数据方面,图数据库较关系型数据库具有更大的优势[14]。本体具有逻辑结构复杂的图式结构,与图数据库的图式结构相符。因此,采用图数据库存储本体可以在最大程度上保持语义信息,更容易表达本体之间的层次结构。另外可以借鉴成熟的部分图算法来对本体进行管理。因此图数据库成为当下存储本体的较好选择。

图数据库基本元素和说明如下:

1) 节点(node)和关系(relation),表现在图模型上就是顶点集合 V 和顶点间边的集合 E。

2) 节点上可以设置属性。

3) 关系上可以设置属性,并且关系的方向可以是单向或双向。

本体的基本建模元语包括:

1) 类(Class):类在本体中表示某个概念或者是某个具体的事务。

2) 关系(Relation):关系在本体中表示概念与概念,事物与事物或者概念与事物之间的相互联系或相互作用。

3) 函数(Function):函数代表事物之间的相互

关系。

4) 公理(Axiom) : 公理就其本身意义来说表达的是真实正确的语句。

5) 实例(Instance) : 实例是类的具体化,类是实例的通用化,表现为对应类的具体事物。

2.2　规则分析

依据本体的建模元语和图数据库的结构特点,给出二者的总体通用映射规则如图 1 所示,其中图 1 右半部分的 2 个节点由上到下分别代表关系的主语和宾语,在图数据库中均表现为节点形式。

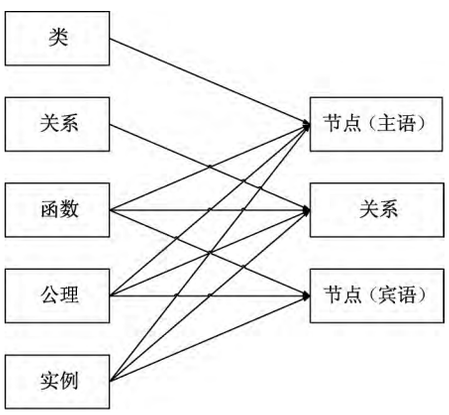

图 1　本体与图数据库存储映射规则图

依据图 1 给出的总体映射规则,对本体的每个建模元语到图数据库的存储映射规则说明如下:

1) 类(Class) 的映射规则

依据类在本体中的含义以及节点在图数据库中的表示含义,可以将本体中的类映射到图数据库中的节点。例如本体中存在一个名为“Wine”的类,将该类映射到图数据库的一个节点,并且给它赋予 name 属性,值为“Wine”。对应类的映射示意如图 2 所示。

图 2　类的映射规则图

2) 关系(Relation) 的映射规则

本体中的关系可以映射为图数据库中的关系,即表示为节点和节点之间的关系,即图的边。

在 wine. owl 中,Wine 类和 Winery 就有 hasMaker 的关系。在这 3 个值中,Wine 为主语,hasMaker 为关系的类型,Winery 为关系的宾语。这样的关系映射到图数据库中,如图 3 所示。

图 3　关系映射模型图

3) 函数(Function) 映射规则

函数包括定义域、值域以及二者的对应规则。目前,本文只考虑一元函数。此时函数的存储映射规则是将函数的定义域和值域分别映射到图数据库的 2 个节点,而它们对应的映射规则映射为图数据库中关于这 2 个节点的一条边,即函数的映射规则类似于关系的映射规则。

4) 公理(Axiom) 映射规则

公理里面包括了本体模型中所蕴含的类关系以及相应的个体关系,用来进一步完善类的定义。例如在“People”这个概念的属性“学习经历”上加一个约束取值范围为“经历”,那么“People”和“经历”就有了一定关系。所以,仍然可以用类似函数的映射规则来存储映射。

5) 实例(Instance) 映射规则

实例(Instance) 依据类的模型而来,通过相应的推导可以发现实例和实例之间的关系以及实例和类模型之间的关系。实例是类的具体化,类是实例的通用化。

在本体到图数据库的映射过程中可以采用分层的思想,其结构如图 4 所示。图的上面一层表示模型层,描述的是本体的类及其之间的相互联系; 下面的一层表示实例层,描述的是类所对应的实例。一个类可能会存在多个实例,一个实例也可能会属于多个不同类。

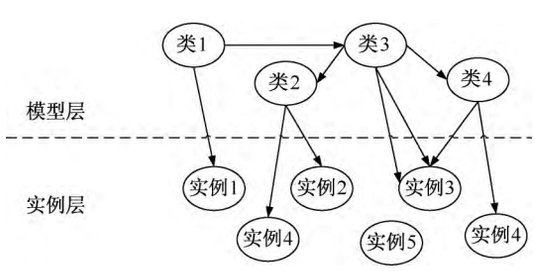

图 4　实例映射模型

按照上述映射规则,可以将本体中的所有建模元语映射为图数据库中的相应元素。

3 算法说明

将本体映射到图数据库，实质共经过两个大的操作：首先对本体进行解析，之后将解析出来的内容映射到图数据库中，其实质是根据本体信息创建一个图。

在解析本体并映射到图数据库中的过程中，类为本体中的基本单元，所以先从遍历类开始。一个类如果不是匿名类，则在图数据库中建立该节点；然后找到该类对应的关系及对应的客体。如果客体不存在，那么便舍弃此关系。因为仅有关系的存在而没有关系所对应的主体和客体存在，或者是只有主体、关系而没有客体存在等，像这样的模式都是没有意义的；如果客体存在，则在图数据库中为此类到对应的客体建立对应的关系。直至本体中所有的类遍历结束。

因为2个类之间可能存在多种对应关系，所以类对应的图中的节点之间可能会存在多条边。

此过程的关键算法如下所示：

1) *for each* (本体中的类 c_i)
2) *if*(c_i不是匿名类)
3) *then* 在图数据库中建立节点 v_i
4) *for each* (c_i对应的每个关系 r_{ij}及该关系对应的客体 c_j)
5) *if* (c_j存在)
6) *then*
7) *if* (v_j在图数据库中不存在)
8) *then* 在图数据库中建立节点 v_j
9) *end if*
10) 在图数据库中为节点 v_i到节点 v_j建立关系 e_{ij}
11) *end if*
12) *end for*
13) *end if*
14) *end for*// 直至其中所有的类遍历结束

4 实验及相关分析

为了验证映射规则和算法的可用性和有效性，本文采用Java语言开发了一个原型系统。该系统采用常见的wine.owl作为实验数据，图数据库选用了较为流行的Neo4j。通过本文的映射规则和算法可以将本体映射到图数据库中进行存储。部分实验结果如表1所示。

表1 wine.owl本体映射的统计数据 个

比较类型	OWL本体	图数据库
类(节点)的数量	137	137
关系的数量	614	614
实例的数量	194	194

从表1可以看出，将本体存储到图数据库中各类元素均保证了相应的完整性。

与其他本体存储方式相比，基于图数据库的本体存储形式可以在保证语义信息完整性的同时，尽量的缩减冗余信息。例如，一个人既是体育明星(类)，又是影视明星(类)。本体建模时为了清晰可能会创建2个实例，分别属于不同的类。而事实上这2个实例就是同一个事物。在本体中这样的情况时有发生，往往会造成信息的冗余。但是如果采用上述映射规则将本体映射到图数据库中，会利用统一资源标识符给每个实例赋属性值，类似于关系数据库中的id，从而保证节点的唯一性。例如 B 和 C 是2个不同的类，一个实体 A 既是 B 的实例，又是 C 的实例。在本体中可能会在 B 和 C 映射的过程中分别给出2个 A，而实际这2个 A 是同一个实例。在图数据库中，则只有一个 A。

Neo4j数据库中，对信息的查询语言为Cypher，它是一种言简意赅的图数据库查询语言，查询模式简单明了，易于理解。本体的查询语言为Sparql，它的查询原理是基于三元组的思想。Cypher可以支持常用的本体查询需求，这就保证了本体存储在图数据库中的查询功能。针对相同的查询需求，各自的查询语句的等价关系如表2所示。

表2 两种查询语句的特点分析

查询需求	Cypher	Sqarql
所有节点和关系	match a-[r]->b return a,r	select ? s ? p ? o where ? s ? p ? o
与wine有disjointWith关系的所有节点	Start a=node:index(name='wine') match a-[disjointWith]->(b) return b	select ? o where 'wine'<http://www.w3.org/2002/07/owl#disjointWith>? o

通过上述实验，可以看出本文提出的本体到图数据库的映射规则和算法，可以实现基于图数据库的本体存储，并仍然支持数据查询，进而证明了这是一种可行的方法。

5 结束语

本文通过对传统本体存储方法进行分析，提出

了一种基于图式的本体存储映射方法。旨在利用二者都具有图结构的特性,避免由于存储结构不匹配而导致的本体语义信息丢失和信息冗余问题。当下很多工作都是针对图数据库和 RDF 数据进行的存储探讨,并没有对本体进行全面讨论。本文给出本体到图数据库的相应映射规则,在最大程度上保持了语义信息的完整性。

本文对本体在图结构领域的应用做了初步探讨,实现了本体到图式结构的存储映射。进一步优化查询效率以及将本体的推理机制有效地应用到图数据库中,将是未来研究工作的重点之一。

参考文献:

[1] Legg C. Ontologies on the semantic Web [J]. Annual Review of Information Science and Technology, 2007, 41(1): 407 - 451.

[2] Neches R, Fikes R E, Finin T, et al. Enabling technology for knowledge Sharing [J]. AI Magazine, 1991, 12(3): 36 - 56.

[3] Gruber T R. A translation approach to portable ontology specifications [J]. Knowledge Acquisition, 1993, 5(2): 199 - 220.

[4] Borst W N. Construction of engineering ontologiesfork nowledge sharing and reuse [J]. UniversiteitTwente, 1997, 18(1): 44 - 57.

[5] Studer R, Benjamins V R, Fensel D. Knowledge engineering: principles and methods [J]. Data and Knowledge Engineering, 1998, 25(97): 161 - 197.

[6] 侯霞. 基于 Topic Maps 的软件工程课程体系模型 [J]. 北京信息科技大学学报, 2012, 27(6): 58 - 61.

[7] Kiryakov A, Ognyanov D, Manov D. OWLIM - a pragmatic semantic repository for OWL [C] // Web Information Systems Engineering-WISE 2005 Workshops. Springer Berlin Heidelberg, 2005: 182 - 192.

[8] Ludwig S A, Rana O F. Performance evaluation of semantic registries: OWLJessKB and instanceStore [J]. Service Oriented Computing and Applications, 2008, 2(1): 41 - 46.

[9] 李勇, 李跃龙. 基于关系数据库存储 OWL 本体的方法研究 [J]. 计算机工程与科学, 2008, 30(7): 105 - 107.

[10] 朱姬凤, 马宗民, 吕艳辉. OWL 本体到关系数据库模式的映射 [J]. 计算机科学, 2008, 35(8): 165 - 169.

[11] 陈磊, 解萍, 吴海波. 基于“属性视图”的 RDB-to-RDF 映射 [J]. 吉林师范大学学报, 2011, 32(2): 67 - 70.

[12] 何向武. 大数据中 RDF 语义数据存储优化探讨 [J]. 计算机应用与软件, 2015, 32(4): 38 - 42.

[13] 项灵辉. 基于图数据库的海量 RDF 数据分布式存储 [D]. 武汉: 武汉科技大学, 2013.

[14] Partner J, Vukotic A, Watt N. Neo4j in action [M]. Virginia of United States: Manning Publications, 2013: 304.

(上接第 58 页)

参考文献:

[1] 周总瑛, 张抗. 中国油田开发现状与前景分析 [J]. 石油勘探与开发, 2004, 31(1): 84 - 87.

[2] 李健, 王月成, 钟权锋. 油井智能间抽采油控制器的研制与应用 [J]. 石油工业计算机应用, 2006(04): 24 - 26.

[3] 吕思平. 油井动液面测量系统的研制 [D]. 北京: 中国石油大学, 2011.

[4] 孙东, 崔晓霖, 齐光峰, 等. 基于动液面连续监测的油井间开优化方法 [J]. 石油石化节能, 2012(08): 1 - 2.

[5] 易其军. 油井动液面自动测量系统研究与设计 [D]. 南昌: 南昌大学, 2014.

[6] 隋立磊. 抽油机井举升高度的重新认识和计算方法 [J]. 内蒙古石油化工, 2012(24): 93 - 95.

[7] 张亮, 白连平. 无线传输式电机参数测量仪的研究 [J]. 北京信息科技大学学报, 2012(05): 86 - 90.