基于 Neo4j 的领域本体存储方法研究*

王 红，张青青†，蔡伟伟，姜 洋
（中国民航大学 计算机科学与技术学院，天津 300300）

摘 要： 在分析民航突发事件应急管理领域本体及其存储特点的基础上，提出了一种基于 Neo4j 的领域本体 RDF 图数据存储方法，研究了领域本体 RDF 有向标记图结构与 Neo4j 图数据库存储模型的关系，结合民航突发事件应急管理领域本体的实例查询，给出了 RDF 图与 Neo4j 之间的映射关系及其实现过程。实验验证了 Neo4j 图数据库在满足领域本体 RDF 图数据查询的同时进一步提高了查询的效率，为大数据平台下的 RDF 图数据语义检索与推理提供了方法支撑。

关键词： 民航突发事件；领域本体；RDF 图数据；Neo4j；映射
中图分类号： TP313.13 **文献标志码：** A **文章编号：** 1001-3695(2017)08-2404-04
doi:10.3969/j.issn.1001-3695.2017.08.038

Research on storage method for domain ontology based on Neo4j

Wang Hong, Zhang Qingqing†, Cai Weiwei, Jiang Yang
(*School of Computer Science & Technoloogy, Civil Aviation University of China, Tianjin* 300300, *China*)

Abstract: Through the analysis of civil aviation emergency management domain ontology and its storage characteristics, this paper proposed a RDF graph data storage method for domain ontology based on Neo4j, and studied the relationship between domain ontology which used the representation of RDF the labled directed graph structure and Neo4j graph database storage model. Combining the instances query for the domain ontology about civil aviation emergency management, it also implemented the process that the mapping relationship between RDF graph and Neo4j. While satisfying the domain ontology RDF graph data query, experiments further improve the efficiency of the query, and provide support method of the semantic retrieval and reasoning of RDF graph data in big data platform.

Key words: civil aviation emergency case; domain ontology; RDF gragh data; Neo4j; mapping

领域本体[1]是指对特定领域内概念及概念间关系的形式化表达，本质上是一种共享的资源描述机制，用于描述一个领域内的知识信息，通常采用 OWL、RDF 等语言描述。领域本体一般采用 OWL 文件或关系型数据库来存储[2~4]。本体数据复杂的图形结构与关系型数据在模型上的不匹配从根本上制约了大规模本体数据的处理能力[5]。基于 HBase 的本体存储[6,7]研究已经取得了一定的效果，但是存储空间消耗太大。

在民航领域中，为了提升民航机场对突发事件的应急处置能力，建立了民航突发事件应急管理领域本体，并对其进行了研究[8~10]。随着信息获取、存储和传播技术的飞速发展，网络多媒体资源日趋复杂化，信息也呈现出多种形态，民航突发事件应急管理对基于大数据的应急决策提出了新要求。

由 Neo Technology 公司提出的 Neo4j[11]是一种高性能的 NoSQL 图数据库，使用图相关的概念来描述数据模型，把数据保存为图中的节点以及节点之间的关系，其具有完全支持 ACID 事务、基于磁盘的持久存储、支持海量数据、高可用的分布式集群、迅速的图查询等特点。本文提出了基于图数据库 Neo4j 的领域本体存储方法，通过领域本体 RDF 图与 Neo4j 之间的映射关系及其实现过程的研究，为民航突发事件应急决策信息的海量数据存储与查询推理提供理论与方法支撑。

1 民航突发事件应急管理领域本体 RDF 存储特点

民航突发事件应急管理领域本体将民航局、各大机场、航空公司与空管局等管理部门的民航突发事件应急预案、案例、规定等非结构化领域知识和各应急单位内部数据库中应急处置相关资源的结构化领域知识有效地组织起来，包括突发事件应急预案、应急案例、应急资源、应急处置规则和应急处置过程等概念及其之间的关系，以 OWL-DL 语言描述。通过解析 OWL 文件，将本体数据表示为 RDF 三元组，而多个 RDF 三元组构成了 RDF 图。对本体数据的存储实质上是对 RDF 图数据的存储[12~14]。

1.1 民航突发事件应急管理领域本体

民航突发事件应急管理领域本体包括类、属性、关系和实例四个元素。

1) 类 指构成突发事件概念的名称，即 RDF 三元组中的主/客体。例如，三元组(应急案例，rdfs:type，Owl:class)表示“应急案例”是一个概念，即类。同理，“应急处置过程”“事故灾害类”“航空器紧急事件”等也是类。

收稿日期：2016-05-24；修回日期：2016-06-29 基金项目：国家自然科学基金委员会与中国民用航空局联合资助项目(U163310052，U1533104)；中央高校基本科研业务费基金资助项目(3122015C022)；中国民航大学科研启动基金资助项目(2014QD14X)

作者简介：王红(1963-)，女，重庆人，教授，主要研究方向为本体技术、数据挖掘、智能信息处理；张青青(1988-)，女(通信作者)，山西人，硕士研究生，主要研究方向为语义网与本体(550569833@qq.com)；蔡伟伟(1989-)，女，河南人，硕士研究生，主要研究方向为语义网与本体；姜洋(1984-)，男，天津人，讲师，主要研究方向为大规模语义数据管理.

2) 关系　描述领域中概念间的相互作用,通过定义域和值域来限定关系的适用范围。其中,定义域为概念集合中的概念,值域可为概念、数据类型、文字。该领域本体概念之间的关系主要包括子类关系(subClassOf) 、参照关系(refer) 和实例与类之间的关系(rdf: type) 等。例如,subClassOf(航空器失事,航空器紧急事件) 表示“航空器失事”是“航空器紧急事件”的子类; refer(MH17,MH17 应急处置过程) 表示 MH17 参照 MH17 应急处置过程; type(MH17,航空器失事) 表示 MH17 的事故类型是属于航空器失事类。

3) 属性　领域本体概念具有两种属性,即对象属性和数据属性。对象属性将对象相互关联,数据属性将对象与数据类型值相关联。如表 1 所示,(“2014-07-17”, rdfs: type, owl: dataProperty) 中的“2014-07-17”是 MH17 的一个数据属性,即 MH17 的发生日期为 2014 年 7 月 17 日,(refer, rdfs: type, owl: objectProperty) 的 refer 是一个对象属性。领域本体概念间的关系也具有属性、描述关系的特性,是对关系的一种约束。

表 1　领域本体属性描述

属性	类型	含义
对象属性	refer/isRefered	参照/被参照
	event_Type	事件类型
	event_Time	事发日期
	event_Place	事发地点
数据属性	event_Cause	事件原因
	event_Level	事件等级
	death_Persons	遇难人数
	…	…

4) 实例　是指实际发生的突发事件,继承突发事件类的属性和关系。例如(MH17, rdfs: type, owl: Namedindividual) 表示马来西亚航空 17 号航班事故实例。不同实例间也具有一些关系, owl: differentFrom 声明了两个个体互不相同, owl: sameAs 声明了两个个体是相同的。例如,(MH17, owl: differentFrom, 911) 表示实例 MH17 和 911 是两个不同类型的事件。

1.2　领域本体的 RDF 数据模型

以 RDF 三元组的集合构成的数据模型为 RDF 图。RDF 三元组〈主体,谓词,客体〉形式化表示为

$$t=\langle s,p,o\rangle\in(I\cup B)\times I\times(I\cup B\cup L)$$

其中: T 为 RDF 三元组的集合,即 $t\in T$; I 表示统一资源标志符(IRIs) ; B 表示空节点(blank nodes) ; L 表示文字(literals) 。

图 1 是以民航突发事件应急管理领域本体为例解析得到的一个 RDF 有向标记图。

说明: 实线椭圆: 本体概念,RDF三元组的主体(subjects)、客体(objects)。
虚线椭圆: 本体属性,RDF三元组的客体(objects)。
有向箭头: 语义关系,RDF三元组的谓词(predicates)。

图1　民航突发事件应急管理领域本体RDF有向标记图

在有向标记图中,同一资源可能既是有向边,又是节点。例如元素 refer 既是 RDF 三元组(refer, rdfs: type, objectProperty) 中的主语,即 RDF 有向标记图中的节点,又是三元组(MH17, refer, MH17 应急处置过程) 中的谓词,即 RDF 有向标记图中的有向边。

2　基于 Neo4j 的 RDF 数据存储方法

2.1　Neo4j 图的数据模型

假设数据类型的集合 $\mathcal{D}$,包含字符串类型 S,即 $S\in\mathcal{D}$, $\mathcal{D}$ 也可包含集合类型的数据类型。对于每一种数据类型 $D\in\mathcal{D}$, dom(D) 表示 D 的值空间,即类型 D 所有可能的值的集,如 dom( ) 表示所有字符串的集。Neo4j 图的形式化定义如下:

定义 1　Neo4j 图又称为属性图(property graph, PG) [15] , $PG=\langle V,E,\mathrm{src},\mathrm{tgt},\mathrm{lbl},\phi\rangle$ 是一个六元组, $\langle V,E,\mathrm{src},\mathrm{tgt},\mathrm{lbl}\rangle$ 表示一个具有标签的有向多重图,其中 V 和 E 分别表示节点和边的集合; 函数 src: $E\to V$ 表示每条边都有起始节点; 函数 tgt: $E\to V$ 表示每条边都有终止节点,函数 lbl: $E\to\mathrm{dom}(S)$ 表示每条边都有标签,函数 ϕ: $V\cup E\to 2^{P}$ 表示属性键值对的集合。

2.2　Neo4j 图数据库存储特点

Neo4j 是一种面向图的数据库,其基元是节点、关系以及属性。节点 Nodes 可以有零到多个属性,这些属性以 key-value 对的形式存在。不同类型的节点通过标签来识别。一个关系 relationship 由一个起始节点 startNode 和一个终止节点 end-Node 构成。与 nodes 一样,关系也可以有多个属性以及标签。

Neo4j 中每一个节点都有一个标签,可以是 iri、literal 或 bnode。iri 的节点具有两个属性,即 kind 和 IRI; bnode 的节点具有一个属性,即 kind、literal 的节点具有四个属性,分别为 kind、value、datatype、language。同一个节点的属性以链表形式存放,如图 2 所示。

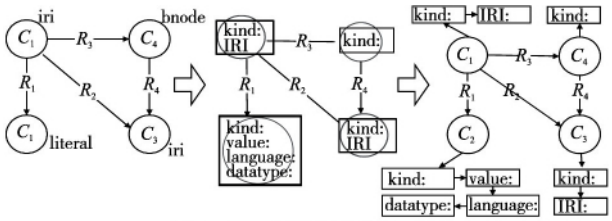

图2　Neo4j图数据库存储过程

2.3　RDF 图与 Neo4j 数据库之间的映射

Neo4j 数据库由节点、关系、节点与关系的属性组成。本体解析后得到 RDF 的节点是属性值。为了方便描述映射过程,引入以下辅助公式:

$$\mathrm{Elems}(t)=\{s,p,o\} \tag{1}$$

$$\mathrm{Elems}(G)=\cup_{t\in G}\mathrm{Elems}(t) \tag{2}$$

$$\mathrm{ttls}(G)=\{t\in G\mid\mathrm{Elems}(t)\cap T\neq\varnothing\} \tag{3}$$

$$\mathrm{SOTerms}(G)=\{x\in(I\cup B\cup L)\mid\langle s,p,o\rangle\in\mathrm{ttls}(G),x\in\{s,o\}\} \tag{4}$$

式(1) 表示所有 RDF 节点对应的属性值的集合,式(3) 表示 RDF 图中三元组的集合,式(4) 表示所有 RDF 的主体和客体对应的属性值的集合。RDF 图与 Neo4j 数据库之间的映射规则定义如下:

a) 节点的集合 V 包含 n_v 个节点, $n_v=|\mathrm{SOTerms}(G)|$,每一个节点都代表一个不同的 RDF 属性值。即存在一个双射函数 v: $\mathrm{SOTerms}(G)\to V$,表示每一个 RDF 属性值 $x\in\mathrm{SOTerms}(G)$ 对应一个不同的节点 $v(x)\in V$。

b) 对于每一个 IRI $u\in I$ 且 $u\in\mathrm{SOTerms}(G)$,属性的集合 $\phi(v(u))=\{\langle\text{"kind","IRI"}\rangle,\langle\text{"IRI",im}(u)\rangle\}$,其中 $v(u)\in V$,单射函数 im(IRI-to-String mapping) 表示 IRI 映射为字符串,即 im: $I\to\mathrm{dom}(s)$ 。

c) 对于每个空节点 $d\in B$, $b\in\mathrm{SOTerms}(G)$,属性的集合

$\phi(v(b))$ = {〈"kind","hash"〉} ,其中 $v(b) \in V$。

d) 对于每个 literal $l \in L$,$L \in$ lSOTerms(G) ,属性的集合为

$$\phi(v(l)) = \{\langle \text{"kind"},\text{"literal"}\rangle, \langle \text{"value"}, vm^{-1}(l)\rangle \langle \text{"datatype"}, \text{im}(\text{dtype}(l))\rangle\} \cup \text{lang}$$

其中:$v(l) \in V$,双射函数 vm(value-to-literal mapping) 表示 value 映射为 literal,vm^{-1} 是其逆函数。即 vm: $v \rightarrow L^*$,$v = \bigcup_{D \in \beta} \text{dom}(D)$,L^* 是 literals 的集合,即 $L^* \subseteq L$ 且 $|L^*| = |v|$。

$$\text{lang} = \begin{cases} \{\langle \text{"language"}, \text{lang}(l)\rangle\} & \text{如果 } l \in \text{dom}(\text{lang}) \\ \varnothing & \text{否则} \end{cases}$$

e) 边的集合 E 包含 n_E 条边,$n_E = |\text{ttls}(G)|$,每一条边表示一个不同的 RDF 三元组 $t \in \text{ttls}(G)$。因此,存在一个双射函数 e: ttls(G) $\rightarrow E$,表示每个 t 对应一条不同的边 $e(t) \in E$。

f) 对于每一个三元组 $t = \langle s, p, o\rangle$,边的标签 $e(t) \in E$ 对应为 im(p) ,且 $e(t)$ 的起始节点和终止节点分别为 $v(s)$ 和 $v(o)$。也即

$$\text{lbl}(e(t)) = \text{im}(p), \text{src}(e(t)) = v(s), \text{tgt}(e(t)) = v(o)$$

g) 对于每个三元组 $t \in \text{ttls}(G)$,边 $e(t) \in E$ 的属性的集合为

$$\phi(e(t)) = \bigcup_{t = \langle s,p,o\rangle, t \in \text{ttls}(G)} \{\langle \text{im}(p), vm^{-1}(o)\rangle\}$$

图 3 以民航突发事件应急管理领域本体中 MH17 为例,给出相应的 RDF 图。

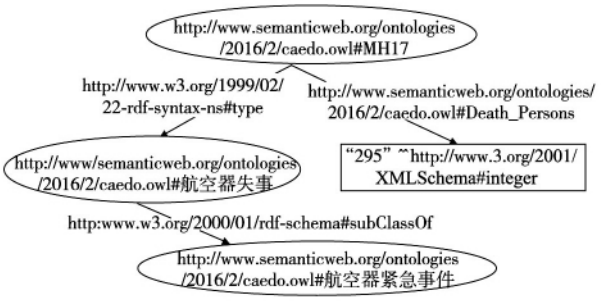

图3 事故案例相关的RDF图

采用 2.3 节中定义的映射规则,得到对应的 Neo4j 图,如图 4 所示,其中 PG 中各个元素分别为

$V = \{v_1, v_2, v_3, v_4\}$,$E = \{e_1, e_2, e_3\}$

$\text{src}(e_1) = v_1$,$\text{tgt}(e_1) = v_2$,$\text{lbl}(e_1)$ = "rdf: type"

$\text{src}(e_2) = v_1$,$\text{tgt}(e_2) = v_3$,$\text{lbl}(e_2)$ = "Death_Persons"

$\text{src}(e_3) = v_2$,$\text{tgt}(e_3) = v_4$,$\text{lbl}(e_3)$ = "rdfs: subClassOf"

$\phi(v_1)$ = {〈"kind","IRI"〉,〈"IRI","MH17"〉}

$\phi(v_2)$ = {〈"kind","IRI"〉,〈"IRI","航空器失事"〉}

$\phi(v_3)$ = {〈"kind","literal"〉,〈"value",295〉,〈"datatype",int〉}

$\phi(v_4)$ = {〈"kind","IRI"〉,〈"IRI","航空器紧急事件"〉}

$\phi(e_i) = \varnothing, i = \{1,2,3\}$

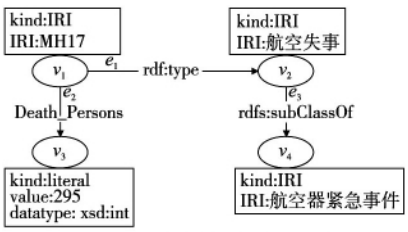

图4 RDF图映射后的Neo4j图

2.4 存储实现

本文主要采用 Jena API 将 RDF 图映射到 Neo4j 图,实现 RDF 图数据到 Neo4j 图数据存储模型的转换。实现过程如图 5 所示。

a) 图 5①所示 RDF 图到③所示 Neo4j 图需采用②所示 Jena API,具体实现的核心代码如下:

```
InputStream in = FileManager. get( ) . open( inputFileName) ;
//创建一个 Jena 模型
Model model = ModelFactory. createDefaultModel( ) ;
model. read( n,"RDF") ( i) ; //读取 RDF 文件
GraphDatabaseService graphab = newGraphDatabaseFactory( )
new EmbeddedDatabase( NEO_STORE) ; //初始化
NeoGraph graph = newNeoGraph( graphab) ; //创建实例 graph
//通过 graph 创建 Jena 模型
Model njmodel = ModelFactory. createModelForGraph( graph) ;
njmodel. add( model) ; //从 model 中将三元组导入到 njmodel
```

b) 图 5③所示为在 Neo4j 提供的基于 node 和 relationship 的索引方式基础上,进一步考虑到本体 properties 的关系,建立了一种针对 nodes、labels、properties 的索引机制,以便提高查询效率。

c) 根据 Neo4j 的文件存储方法,图 5④给出了 nodes(包括 neostore. nodestore. db、neostore. nodestore. db. labels 等) 、relationships(包括 neostore. relationshipstore. db、neostore. relationshiptypestore. db 等) 、properties 存储文件(包括 neostore. propertystore. db 等) 。

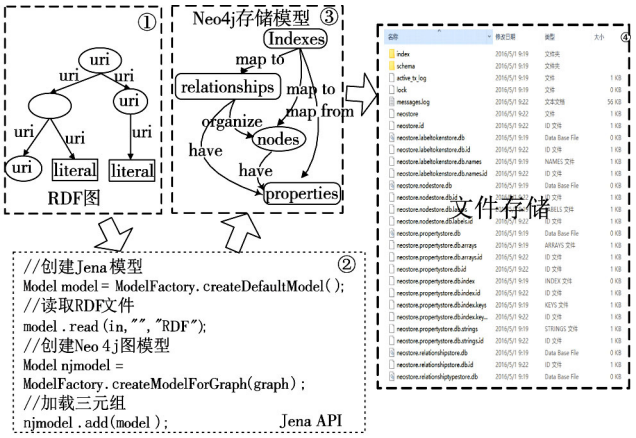

图5 RDF到Neo4j的存储实现过程

3 基于图数据的查询

3.1 Neo4j 查询转换

SPARQL 是目前常用的本体查询语言,其指定一组 RDF 三元组匹配模式,根据这组三元组匹配模式从 RDF 数据集中搜索匹配的 RDF;但是目前的 Neo4j 图数据库没有提供 SPARQL 的查询接口,需要将 SPARQL 查询语言转换为 Neo4j 支持的数据查询语言 CYPHER,实现的算法如下。

算法描述:

```
Function Spargl2Cypher( sparglString)
Iuput: sparglString //spargl 查询语句
Output: cypherString //cypher 查询语句
begin
cypherString 初始化为空集
query←QueryFactory. create( sparqlString)
op←Algebra. compile( query)
//采用 JenaARQ 解析 SPARQL 语句
afterParse = new String( op. toString( ) )
brk←afterParse. spilt( ? triple")
//从 select 部分提取出将返回结果值的变量集合
RetVarSet←brk [0]
for i←1 to brk. length - 1 do
TriplepatternSet [i - 1] ←brk [i]
end for
//用邻接表来存储三元组
AdjacencyList( TriTriplePatternSet,brk. length - 1)
//深度优先遍历邻接表,记录所有的遍历路径
pathSet←DFSTraverse( )
```

```
/* 记录非变量的集合,
每一条遍历路径对应 match 的一个分句* /
for path: pathSet do
ConditionNodeSet. add( path)
matchString←path
end for
//从 ConditionNodeSet 中任选一个顶点,作为 start 的起始顶点
startString←ConditionNodeSet
returnString←RetVarSet
cypherString←startString + matchString + returnString
return cypherString
end
```

3.2　实现过程与效果

以图 1 所示的民航突发事件应急管理领域本体 RDF 有向标记图为例,为了查询“MH17”及其相关的节点与关系,根据 3.1 节的算法设计了表 2 的查询转换。

表 2　SPARQL 与 CYPHER 查询的转换

查询方式	查询语句
SPARQL (sparqlString)	PREFIX: 〈http: //www. semanticweb. org/ontologies/2016/2/caedo. owl#〉 PREFIX: rdf 〈http: //www. w3. org/1999/02/22-rdf-syntax-ns#〉 SELECT ? x ? y ? z ? u ? v ? w ? m WHERE{ MH17 : event_Time ? x. MH17 : event_Cause ? y. MH17 : death_Person ? z. MH17 : Refer ? u. MH17 rdf: type ? v. MH17 rdfs: type ? w. MH17 Owl: differentFrom ? t. } LIMIT 6
CYPHER (cypherString)	START m = node: case(IRI = ‘MH17’) MATCH path = (m) - [] →( ) RETURN path LIMIT 6

查询结果为:

```
result{
    { “MH17”,: event_Time, “2014-07-17”} ,
    { “MH17”,: event_Cause, “导弹击落”} ,
    { “MH17”,: death_Persons,295 } ,
    { “MH17”,: refer, “MH17 应急处置过程”} ,
    { “MH17”,: rdf: type, “航空器失事”} ,
    { “MH17”,: owl: differentFrom, “911”}
  }
```

查询结果表示,节点“MH17”与节点“2014-07-17”之间存在 event_Time 关系,节点“MH17”与节点“导弹击落”之间存在 event_Cause 关系,其他类似。为了进一步验证该存储方案的查询效率,本文采用了 Berlin SPARQL Benchmark(BSBM) [16] 标准数据集进行测试,利用 Neo4j Cypher Java API 执行 CQL 命令来执行查询,查询过程的关键代码如下:

```
ExcutionEngin execEngine = new ExcutonEngine( graphab) ;
Sparql2Cypher queryCypher = new Sparql2Cypher( ) ;
String query = queryCypher. Sparql2Cypher( sparqlString) ;
ExecutionResult execResult = execEngine. execute( query)
```

实验结果如表 3 所示。其中的 D_1、D_2、D_3、D_4、D_5 分别为 14 MB、71 MB、284 MB、1.4 GB、7 GB 的数据集。

表 3　不同数据集及其响应时间

数据集	三元组数量/条	响应时间/ms	
		OWL	Neo4j
D_1	50 116	2 665	53
D_2	250 492	3 724	89
D_3	1 000 226	8 714	181
D_4	5 000 453	9 891	678
D_5	25 000 55	14 044	4 293

Neo4j 数据库重点解决了拥有大量连接的传统 RDBMS 在查询时出现的性能衰退问题。通过围绕图进行数据建模,Neo4j 会以相同的速度遍历节点与边,其遍历速度与构成图的数据量没有任何关系。平均响应时间如图 6 所示。由图 6 可以看出,基于 Neo4j 图数据库的查询效率明显高于 OWL 文件式查询,验证了该存储方案的有效性。

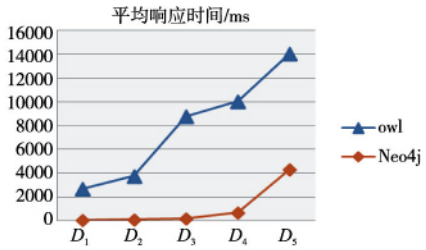

图6　平均响应时间

4　结束语

民航应急管理作为航空安全管理的重要组成部分,应急决策过程的基础信息与领域知识具有数据结构复杂、数据规模增长迅速的特点。本文针对目前民航突发事件应急管理领域本体图数据存储与应用中存在的问题,提出了一种基于 Neo4j 的领域本体图数据存储方法,实现了领域本体与 Neo4j 图数据库之间的映射。通过实验分析了该存储方案的有效性,为下一步深入研究基于领域本体 RDF 图数据的语义检索与推理奠定了良好的数据与方法支持。

参考文献:

[1] Bozsak E, Ehrig M, Handschuh S, *et al*. KAON-towards a large scale semantic Web [C] //Proc of E-commerce and Web Technologies. Heidelberg: Springer, 2002: 304-313.

[2] Wang Hong, Zhu Yueli, Wang Jing. The applied research of the method in ontology mapping based on the relational mode [J]. Journal of Convergence Information Technology, 2013, 18(11) : 23-32.

[3] Das S, Chong E I, Eadon G, *et al*. Supporting ontology-based semantic matching in RDBMS [C] //Proc of the 13th International Conference on Very Large Data Bases. 2004: 1054-1065.

[4] Pan Zhengxiang, Heflin J. DLDB: extending relational database to support semantic Web queries [C] //Proc of the 1st PASS. Santa Barbara: Informal Proceedings, 2003: 43-48.

[5] 邹磊,陈跃国. 海量 RDF 数据管理 [J]. 中国计算机学会通讯, 2012, 11(8) : 32-43.

[6] 赵磊. 基于 HBase 的本体存储与查询的研究 [D]. 南昌: 华东交通大学, 2011.

[7] 朱敏,程佳,柏文阳. 一种基于 HBase 的 RDF 数据存储模型 [J]. 计算机研究与发展, 2013, 50(S1) : 23-31.

[8] 周文涛,王红,王静,等. 民航应急决策方案语义模型构建方法的研究 [J]. 计算机应用研究, 2013, 30(1) : 195-198, 210.

[9] 王红,杨璇. 民航突发事件应急管理领域本体构建方法的研究与实现 [D]. 天津: 中国民航大学, 2010.

[10] 王红,高斯婷,潘振杰,等. 基于 NNV 关联规则的非分类关系提取方法及其应用研究 [J]. 计算机应用研究, 2012, 29(10) : 3665-3668.

[11] Neo4j [EB/OL]. (2016-05-17) [2016-05-21]. http: //neo4j. org/.

[12] Bonstrom V, Hinze A, Schweppe H. Storing RDF as a graph [C] // Proc of the 1st Conference on Latin American Web Congress. 2003: 27-36.

[13] 项灵辉,顾进广,吴钢. 基于图数据库的 RDF 数据分布式存储 [J]. 计算机应用与软件, 2014, 31(11) : 35-39.

[14] 康杰华,罗章璇. 基于图形数据库 Neo4j 的 RDF 数据存储研究 [J]. 信息技术, 2015(6) : 115-117.

[15] Hartig O. Reconciliation of RDF* and property graphs [R]. [S. l.]: University of Waterloo, 2014.

[16] Bizer C, Schultz A. The Berlin SPARQL benchmark [J]. International Journal on Semantic Web and Information Systems, 2009, 5 (2) : 1-24.