International Journal of Advanced Computer Technology (IJACT)

ISSN:2319-7900

RESEARCH ON DISTRIBUTED REASONING METHOD OF DOMAIN ONTOLOGY BASED ON RDFS

Hong Wang, School of Computer Science and Technology, Civil Aviation University of China, Tianjin 300300, China;
Weiwei Cai, School of Computer Science and Technology, Civil Aviation University of China, Tianjin 300300, China;
Xiaodan Guo, School of Computer Science and Technology, Civil Aviation University of China, Tianjin 300300, China;
Yang Jiang,School of Computer Science and Technology, Civil Aviation University of China, Tianjin 300300, China

Abstract

To address the problem of distributed reasoning in the civil aviation emergency management domain ontology, on the basis of the analysis of the application characteristics of RDF graph data and RDFS reasoning rules for domain ontology, this paper adopts a distributed semantic data processing method by separating domain ontology model data and instance data. The RDFS rules matching algorithm of domain ontology model data and instance data based on Hadoop and the implementation of the Map and Reduce processing are presented.The experimental results verify the distributed reasoning algorithm for large-scale domain ontology is efficiency, which provides a good data and method support for civil aviation emergency causal reasoning and correlation analysis based on RDF graph data.

Introduction

The civil aviation emergency management domain ontology[1][2] refers to the civil aviation related concepts and formal expression, mainly including emergency plan, emergency cases, emergency disposal and emergency resources and other information, which are defined with triple format. Domain ontology is generally described in OWL(Web Ontology Language) or RDF[3][4] (Resource Description Framework) language.With the continuous development of the Semantic Web, the semantic data,which involved in emergency decision-making of civil aviation[5], showing the features of large data volume and rapid growth. The depth mining of the large-scale semantic data is very important to improve the ability of the civil aviation emergency decision-making. At present, the ontology reasoning is mainly based on Jena[6]and Pellet[7], etc,which can't deal with distributed reasoning of large-scale ontology data.In this paper, we will combine the distributed processing of MapReduce[8-14] with RDFS[15] reasoning of domain ontology, and further research on the distributed reasoning methods of large-scale domain ontology, and provide the method support for the emergency management and collaborative decision making in large data environment.

Related Research

A. Domain Ontology RDF Graph Data

CAEDO[16](Civil Aviation Emergency Domain Ontology）,is defined as a triple, and denoted as CAEDO = (C, R, I),among them,C is the set of relevant concepts, R is the set of relations between concepts and concepts, concepts and attributes, I is the instance of the classes.

An RDF triple mainly consists of subject, predicate, and object, and is formally defined as:

$t=(s,p,o)\in(U\cup B)\times(U\cup B)\times(U\cup B\cup L)$

Where: s, p, o respectively represent the subject set, predicate set and object set, U represents the set of Uniform Resource Identifier (URL), B represents the set of anonymous nodes, and L represents the set of any type texts.

The data model of RDF is the RDF graph, which is expressed as the directed label graph. Figure 1 shows the RDF directed graph of the mapping relationship between concepts and instances ,that described in the civil aviation emergency management domain ontology.

RDFS (RDF Schema) is used to limit the application fields and constraints of RDF data, which not only contains the RDF vocabulary, but also adds custom words, including the class, the attributes and so on.

The semantic data of civil aviation emergency management domain ontology is composed of two parts: pattern data and instance data:

Pattern data refers to the relationship of the concepts ,like emergency cases and emergency plan, which are described by RDFS and OWL modeling languages on domain ontology.

Instance data refers to the semantic concepts model based on the definition, with the explicit semantic, used to describe instance triple of the specific thing , that is instance triples,such as,(MH17, rdfs: type, Owl: Namedindividual), which represents an instance of number 17 Malaysia Airlines flight accident.

As shown in Figure 2,it is an RDFS example of the civil aviation emergency management domain ontology, what above the dotted line for the RDF model data, and below the dotted line for the RDF instance data.

RDF reasoning is based on the two parts of semantic data,which from the ontology to infer the relevant new semantic information.For example, in Figure 2,we can get a new triple(MH17, rdf: type, aircraft emergency),according to triples (aircraft crash, rdfs:subClassOf, aircraft emergency) and (MH17, rdf: type, aircraft crash).

B. Analysis of RDFS reasoning rules

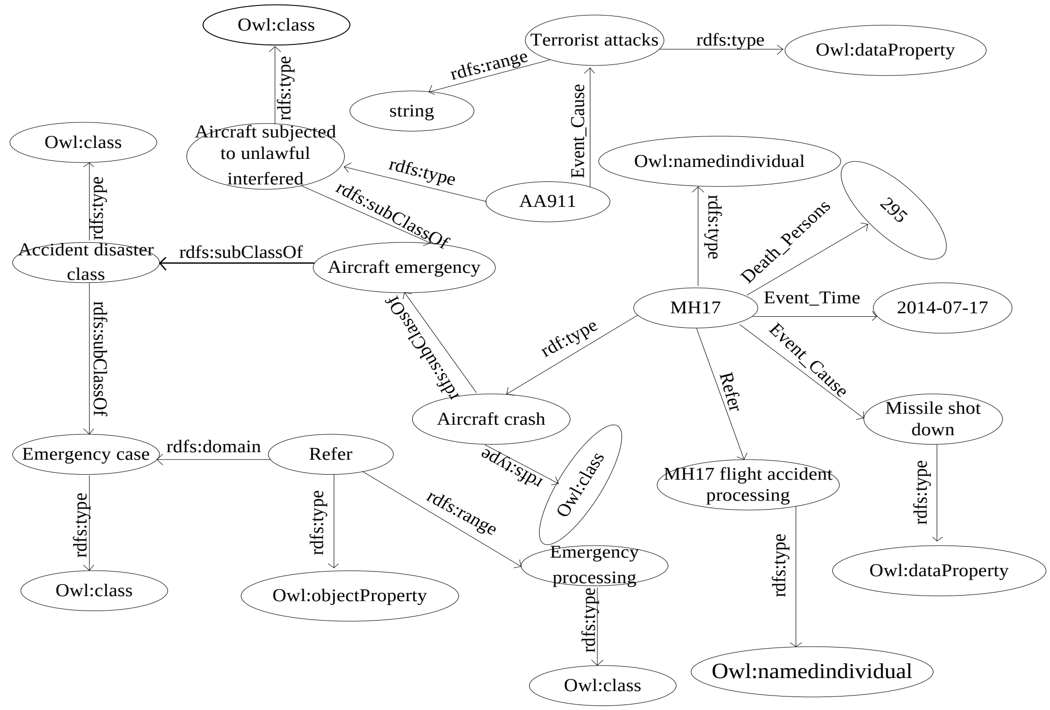


Figure1. The Cvil Aviation Emergency Management Domain Ontology RDF Graph

The reasoning of large-scale domain ontology is essentially a iteration process with the RDFS rules,RDFS includes 14 reasoning rules [17],among them, the rules 2, 3, 5, 7, 9 and 11 will produce the most part of triples, and there are some relationships between the inputs and outputs of these rules. We will discuss most about these rules. We first categorise the rules based on their input:

1) rule 2,3 and 7 have a instance triple at least.

2) rules 5 and 7 have a triple with rdfs:subPropertyOf as predicate at least.

3) rules 9 has a triple with rdf:type as predicate at least.

4) rules 9 and 11 have a triple with rdfs:subClassOf as predicate at least.

Based on their output:

1) rules 5 produces schema triples with rdfs:subPropertyOf as predicate,

2) rules 11 produces schema triples with rdfs:subClassOf as predicate,

3) rules 2, 3,and 9 produce instance triples with rdf:type as predicate,

4) rule 7 may produce arbitrary triples.

Figure 3 displays the dependencies between the RDFS rules, connecting rules based on their input and output.

As shown in Figure 3, we can find out that the output of the rule 5 can serve as the input of the rule 7,and the output of rule 7 can serve as the input of rule 2 and 3,and the output of rule 3 and 11 can serve as the input of rule 9. we can firstly reasoning on the high priority rules, and the reasoning results can serve as the input of low priority rules.

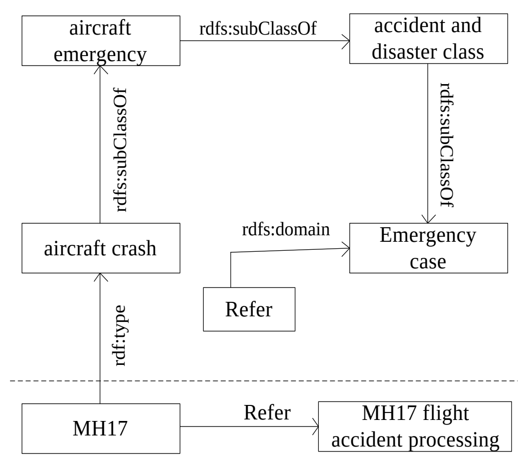


Figure 2. An RDFS Example of Domain Ontology

ISSN:2319-7900

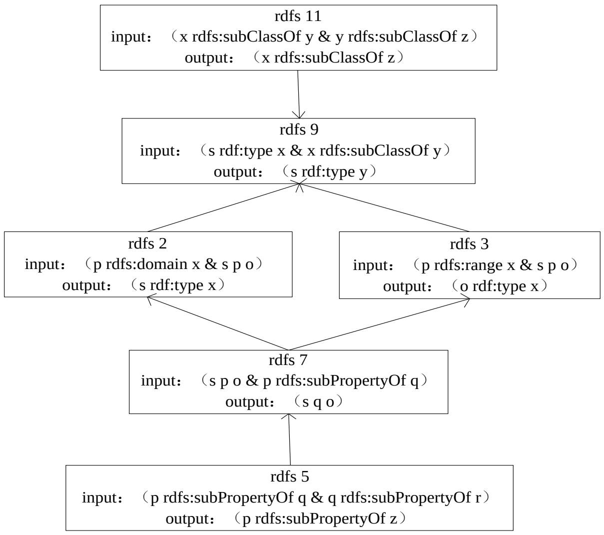


Figure 3. RDFS Rule Dependencies

Distributed Reasoning Based on RDFS

The RDFS distributed reasoning process of domain ontology adopts the method of separate schema data from the instance data processing. Through RDFS rule matching algorithm design to achieve the Map and Reduce processing. The research idea as shown in Figure 4:

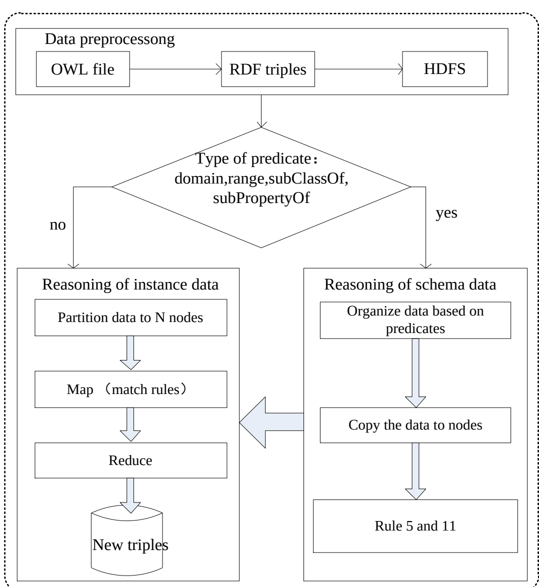
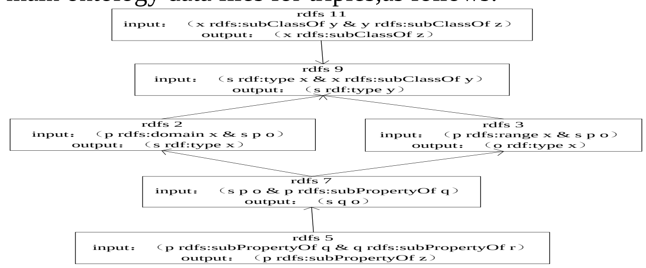


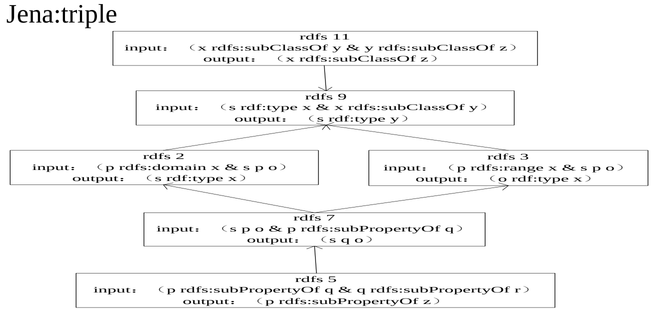
Figure 4. RDFS Distributed Reasoning Process of Domain Ontology

A. Data Preprocessing

Domain ontology data is generally stored via the OWL file. Therefore, we should firstly use Jena to parse the domain ontology data files for triples,as follows:



We can parse the files as the following through files the Jena:triple



The RDF data is separated schema data from instance data according to the predicate types, and stored in HDFS as s file respectively. The schema data is directly copied to each node of the local memory as the global data, and the M instance data is equally divided into N cluster nodes to parallel processing. That is, M / N instance data is stored on each node by key-value pairs, where key means predicate, value means a triple.

B. Schema Data Reasoning

The reasoning of schema data is a process of rule matching with rule 5 and 11, the purpose is to provide the basis for the reasoning of instance data in the Map phase.The process is based on the various predicates to organized schema data together,the triples that match with rule 2 (rdfs: domain) are stored in the domainData, and the triples that match with rule 3 (rdfs: range) are stored in the rangeData ,and the triples that match with rules 5 and 7 (rdfs: subPropertyOf) are stored in the subPropertyOfData, and the triples that match with rules 9 and 11 (rdfs: subClasssOf) are stored in subClassOfData.On this basis, we can use the rule 5 and 11 to realize the reasoning of schema data. The algorithm is as follows:

Algorithm 1:

```
Setup(key,value)
{
  while(value.next()!=null)
  {
      if(value.predicate=="rdfs:domain")
          domainData.add(new triple(value.subject,value,object))
      else if(value.predicate=="rdfs:range")
          rangeData.add(new triple(value.subject,value,object))
      else if(value.predicate=="rdfs:subClassOf")
          subClassOfData.add(new triple(value.subject,value,object))
      else if(value.predicate=="rdfs:subPropertyOf")
          subPropertyOfData.add(new triple(value.subject,value,object))
   }
   subClassOfData.transitiveClosure();
   subPropertyOfData.transitiveClosure();
}
reduce();
```

C. Instance Data Reasoning

a.Map processing

In the Map method, do the rules matching for M / N instance data on each node,On the Map side, from the HDFS file to read key-value pairs of instance data.To read in the instance data of domain ontology according to rules 2, 3, 7,and 9 matching and reasoning, the specific process is as follows:

To read the triples of instance data to judge. In algorithm 1, if all of the subproperty sets (subperproperties) contain the predicates of the instance triples by rule 5 reasoning , and output the key-value pairs of instance triples, then the output of rule 5 with matching instance triples by rule 7 reasoning ,and output all key-value pairs of triples, and the output of triples are satisfied with rules 2 and 3.

To read the triples of instance data and the the results of reasoning form step 1 to judge. If the domainData contains the predicate of instance triples, firstly,output the key-value pairs of the instance triples,and then match with rule 2,through loop to get the triples of all types of triples object.Output the key-value pairs of triples.

To read the triples of instance data and the the results of reasoning form step 1 to judge. If the rangeData contains the predicate of instance triples, firstly,output the key-value pairs of the instance triples,and then match with rule 3,through loop to get the triples of all types of triples object.Output the key-value pairs of triples.

When the predicate of the triples is type, output the key-value pairs of triples,and combine the subclasses sets form algorithm 1 reasoning to complete the match with rule 9, and output the key-value pairs of triples.The rule 9 is the top of the rule dependency, thus the reasoning of ontology is completed.

The algorithm is as follows:

Algorithm 2:

```
Map(key,value)
key:triples
value:new triples
if(subproperties.containsKey(value.predicate))     //rdfs7
   emit(value.predicate,{triple})
   superproperties.add(superproperties.get(value.object))
for(superproperty in superproperties)
   emit(superproperty,{triple})
   newtriple={value.subject,superproperty,value.object}
   rule2(newtriple)                                  //rdfs2
if(ranges.containsKey(value.predicate))
  for(type)  emit("rdf:type",{triple})
  rule3(newtriple)                                   //rdfs3
if(range.containsKey(value.predicate))
  for(type) emit("rdf:type",{triple})
else if(value.predicate=="rdf:type")                 //rdfs9
  for(superclass) emit("rdf:type",{value.subject,class})
emit
```

Based on the results of schema data reasoning, adopt the algorithm 2 to realize the reasoning of instance data. The reasoning process is as follows:

Adopt the rule 2 and 3 to matching: the instance triples (MH17, Refer, MH17 flight accident processing) and schema triples (Refer, rdfs: domain, emergency case) ,(Refer, rdfs: range, emergency disposal)match the rule 2 and 3 respectively, and trigger the rule 2 and 3 to get the key-value pairs of new triples:("rdf: type",{MH17,rdf:type,emergency case})and("rdf: type",{MH17 flight accident processing, rdf:type, emergency disposal}).

Adopt the rule 9 to matching: Firstly, the resoults of schema triple reasoning (aircraft being unlawful interfered, rdfs: subClassOf, accident and disaster class) and (AA911, rdf:type, aircraft being unlawful interfered) match the rule 9 to get the new triple(“rdfs: type”, {A911,rdf:type, accident and disaster class), then the result of reasoning from step 1(MH17,rdf: type, emergency case) as the output with the schema triple (Aircraft crash, rdfs:subClassOf, emergency case) match the rule 9 to get the new triple（“rdf:type”, {MH17, rdf:type, accident and disaster class}） and （“rdf:type”, {MH17, rdf:type, aircraft emergency}).

b.Reduce processing

In reduce method, we just reducing the duplicates of triples.The processing is as fllows:in the Map phase, the triples of reasoning are key-value pairs, so we can through the value to judge. If the value is unique, there is no duplicates of triples, if not,the key-value pairs of original triple will be deleted.The algorithm is as follows:

Algorithm 3:

ISSN:2319-7900

$$\begin{aligned}&\text{Reduce}(\text{key}, \text{value\_list})\\&\text{Key}: \text{triple.predicate}\\&\text{value\_list}: \{\text{triple}\ \}\\&\text{for}(\text{triple in value\_list})\\&\text{if }(\{\text{triple} == \text{false})\\&\text{remove}(\text{key}, \text{value})\\&\text{end}\end{aligned}$$

In the figure 6, the process of schema reasoning by:

("rdfs:subClassOf", { aircraft being unlawful interfered, rdfs: subClassOf, accident and disaster class}).

("rdfs:subClassOf", {accident and disaster class, rdfs: subClassOf, emergency case}).

The reasoning results of triples:

("rdfs:subClassOf", {aircraft being unlawful interfered, rdfs: subClassOf, emergency case}).

As well as:

("rdfs:subClassOf", {aircraft being unlawful interfered, rdfs: subClassOf, aircraft emergency}).

("rdfs:subClassOf", {aircraft emergency,rdfs: subClassOf, emergency case}).

The reasoning results of triples:

("rdfs:subClassOf",{aircraft being unlawful interfered, rdfs: subClassOf, emergency case}).

Due to the reasoning results of triples exist the duplicates of triples, so we should combine data with the same value into (key, value_list). This process is the Reduce method.

D. Results and analysis

The experiment was performed on a Hadoop cluster with 8 nodes, Hadoop version 2.4.1, the main hardware configuration of cluster are 4GB memory, CPU for the Intel Core I7-3770,1TB mechanical disk. The operating system is CentOS 7.0.1406. The development environment is JDK 1.7.0_72.

The paper uses the civil aviation emergency management domain ontology by Civil Aviation University of China as the experimental data. The data mainly consists of 289 classes, 1374 attributes and 6138 instances.

In order to test the efficiency of our method, the comparison is made with the advanced RDF reasoning method WebPIE.When dealt with the different data volume, the comparison of experimental data is shown in Table 1.

Table 1 Different Data Volume and Reasoning Time (s)

Data set	Number of triples	Reasoning/s	
		Our method	WebPIE
D1	6280000	38.34s	69.43s
D2	15784000	93.24s	170.48s
D3	37984200	134.57s	258.89s

Where D1, D2, D3 represent respectively1.7G, 4.25G, 10.2G data sets.

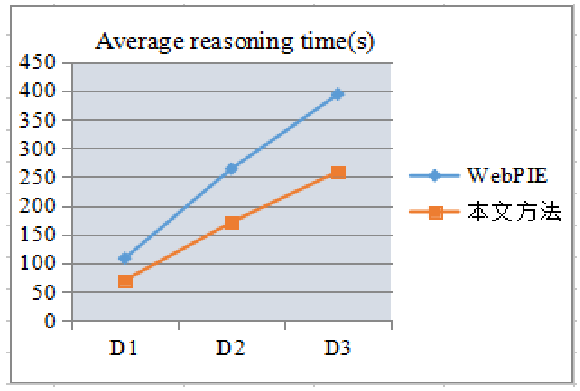


Figure 5. Average Reasoning Time

It can be seen from Figure 5 that the reasoning efficiency of our method is significantly higher than WebPIE. The validity of the reasoning scheme is verified.

a.Reasoning results of schema data

According to algorithm 1, the figure 1 shows the RDF graph of domain ontology,which the process of schema data reasoning is as follows:

1) According to the predicate of triples to judge,output all key-value pairs of schema triples to matching rule 5 and 11,namely:

("rdfs:subClassOf", {aircraft emergency, rdfs: subClassOf, accident disaster class}),

("rdfs:subClassOf", {accident disaster class, rdfs: subClassOf, emergency case}),

("rdfs:subClassOf", { aircraft being unlawful interfered, rdfs: subClassOf, aircraft emergency}),

("rdfs:subClassOf", {aircraft crash, rdfs: subClassOf, aircraft emergency}).

2) outputs from step 1 are matched rule 11, and output all the key-value pairs of triples.

In Figure 6,the dashed arrows show the reasoning results of schema data,namely:

("rdfs:subClassOf", {aircraft crash, rdfs: subClassOf, emergency case}),

("rdfs:subClassOf", {aircraft crash, rdfs: subClassOf, accident disaster class}),

("rdfs:subClassOf", {aircraft emergency, rdfs: subClassOf, emergency case}),

("rdfs:subClassOf", {aircraft being unlawful interfered, rdfs:subClassOf, accident disaster class}),

("rdfs:subClassOf", {aircraft being unlawful interfered, rdfs: subClassOf, emergency case}).

b.Reasoning results of instance data

ISSN:2319-7900

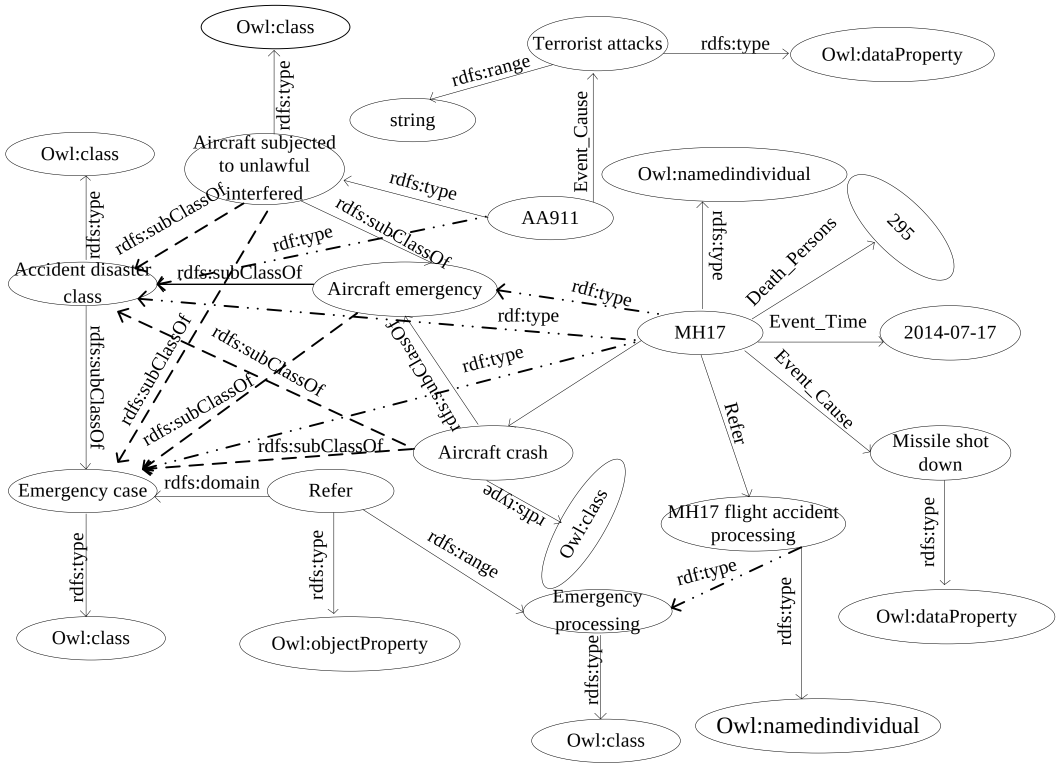


Figure 6. Show the Rreasoning Results

According to algorithm 2, in Figure 6,the solid line arrows show the reasoning results of instance data,namely:

("rdf:type", {MH17, rdf: type, emergency case}),

("rdf:type",{ MH17 flight accident processing, rdf: type, emergency processing}),

("rdf:type", {AA911, rdf: type, accident disaster class}),

("rdf:type", {MH17, rdf: type, accident disaster class}),

("rdf:type", {MH17, rdf: type, aircraft emergency}).

For the RDFS distributed reasoning in Figure 1, which get the 10 new semantic relations. From the RDFS reasoning results above ontology,we can analyse both MH17 and AA911 are the instances of aircraft emergency,and due to AA911 instance belong to terrorist incident,so MH17 is also belonged to the instance of terrorist incident.

Conclusion

In this paper, mainly for the problem of depth mining of large-scale semantic data of the civil aviation emergency management domain ontology, which combine the MapReduce method on Hadoop with RDFS reasoning of domain ontology, and achieve the reasoning of the large data environment of domain ontology. The next, we will combine the transmission mechanism of concept relations of domain ontology, and further research on causal reasoning methods of Civil Aviation Emergency,and and provide methods support for civil aviation emergency management and collaborative decision-making.

Acknowledgement

This work is partially supported by the Joint Funds of the National Natural Science Foundation of China and the Civil Aviation Administration of China (U1633110).

References

[1] Hong Wng,Xuan Yang. Research on Ontology-Based Knowledge Presentation and Reasoning in Civil Aviation Emergency Decision [D]. Tian jin: Civil Aviation University of China.

[2] Man Li,Dazhi Wang. Dynamic Composition of Web Services Based on Domain Ontology[J]. Chinese journal of computers, 2005,28(4):644-650.

[3] Shanping Li,Yujie Hu,etc. Overview of Researches on Ontology[J].Journal of computer research and development, 2004, 41(7):1041-1052.

[4] Jingfen Wang,Zhili Fang. Distributed Optimized Query Algorithm Based on Index[J].Computer science, 2014, 41(11):233-238.

[5] Hong Wang,Xuan Yang. Research on Ontology-Based Knowledge Presentation and Reasoning in Civil Avia-

ISSN:2319-7900

tion Emergency Decision[J].Computer engineering and science. 2011, 33(4):129-133.

[6] Mcbride B. Jena:A semantic Web toolkit[J]. IEEE Internet Computing, 2002, 6(6):55-59.

[7] Sirin E, Parsia B, Bernardo E C, et al. Pelet:A practical OWL—DL[J]. Web Semantics:Science, Services and Agent on the World Wide Web, 2007, 5(2):51-53.

[8] Husain M F, Khan L, Kantarcioglu M, et al. Data Intensive Query Processing for Large RDF Graphs Using Cloud Computing Tools[C]// 2010 IEEE 3rd International Conference on Cloud Computing.IEEE Computer Society, 2010:1-10.

[9] White T. Hadoop : the definitive guide[J]. O'reilly Media Inc Gravenstein Highway North, 2010, 215(11):1 -4.

[10] Yizhi Wang,Weidong Sun. Key Technologies of Distributed Storage for Cloud Computing[J].Journal of software, 2012, 23(4):962-986.

[11] Rong Gu,Fangfang Wang,Chunfeng Yuan. YARM: Efficient and Scalable Semantic Reasoning Engine Based on MapReduce[J].Chinese journal of computers, 2015, 38(1):74-85.

[12] Shixing Wang. Research of Large-scale RDF Graph Parallel Reasoning Method Based on the MapReduce[D].Nanchang Uinversity,2014.

[13] Jang B, Ha Y G. Transitivity Reasoning for RDF Ontology with Iterative MapReduce[C]// Seventh International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing. IEEE, 2013:232-237.

[14] Zhenxin Qu,Chuanming Yu. Inference of mass ontology based on cloud computing[J].Journal of computer applications, 2011, 31(12):3324-3326.

[15] Cetin Y, Abul O. Distributed RDFS Reasoning with Map Reduce[M]. Springer International Publishing, 2014.

[16] Jian Li.Application Study of Method for Fuzzy Extension in Civil Aviation Emergency Domain Ontology[D]. Tian jin: Civil Aviation University of China.

[17] Urbani J, Kotoulas S, Oren E, et al. Scalable Distributed Reasoning Using MapReduce[M]//The Semantic Web-ISWC 2009.2009:634-649.

Biographies

HONG WANG born in 1963, M. S, professor, CCF member. Her research interests include ontology technology, data mining and intelligent information processin.

WEIWEI CAI born in 1989, M. S. candidate. Her research interests include semantic web and ontology.Miss Cai may be reached at 740146473@qq.com