Study on Query System Based on Pomology Domain Ontology

Qian Sun, Qiulan Wu, Yong Liang

▶ **To cite this version:**

Qian Sun, Qiulan Wu, Yong Liang. Study on Query System Based on Pomology Domain Ontology. 5th Computer and Computing Technologies in Agriculture (CCTA), Oct 2011, Beijing, China. pp.180-187, 10.1007/978-3-642-27281-3_23 . hal-01351809

HAL Id: hal-01351809

https://inria.hal.science/hal-01351809

Submitted on 4 Aug 2016

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Distributed under a Creative Commons Attribution 4.0 International License

Study on Query System Based on Pomology Domain Ontology

Qian Sun [1],Qiulan Wu[1], Yong Liang[1]

[1] School of Information Science and Engineering,
Shandong Agricultural University,
Taian, China, 271018
E-mail:applesq@163.com

Abstract. This paper studied the construction of Pomology Domain Ontology (PDO), and the realization of PDO-based query system. First, an approach to build PDO based on Agriculture Science Thesaurus (AST) was proposed, which consists of confirming core concepts, adding the properties of concepts, confirming the relationships between concepts, adding the instances of concepts, and representing domain ontology. Then the PDO-based query system model and implementation algorithm were given. The query system realized class query, instance query, and property query by Jena. Query results indicate that the algorithm is practical and the search time is shortened. Through the query system pomology knowledge can be obtained from PDO according to user needs.

Keywords: Protégé; Jena; query; owl; Pomology; ontology

1 Introduction

With the development of owl ontology language, more and more knowledge systems based on domain ontology are developed. The development of knowledge system includes knowledge representation, storage, reasoning, query and so on, in which query technique is one of the key technologies, through it knowledge can be obtained from ontology [1]. In order to realize pomology knowledge query from PDO according to several conditions demanded by users, the query system based on PDO is studied in this paper.

2 Tools on Ontology

In recent years, a variety of tools have been developed by different research institutes, including ontology editing tools, ontology parsing tools and so on. Protégé and Jena are used to model and parse PDO in this study.

2.1 Protégé

Protégé is a free, open source ontology editor, which allows developers to model ontology. The Protégé platform supports two main ways of modeling ontology via the protégé-frames and protégé-owl editors. Protégé ontology can be exported into a variety of formats including RDF(S) [2], OWL [3], and XML Schema. Protégé can be customized to provide domain-friendly support for creating knowledge models. Further more, Protégé can be extended by way of a plug-in architecture and a Java-based Application Programming Interface (API) for building knowledge-based tools [4].

2.2 Jena

Jena is a Java framework for building Semantic Web applications. It provides a programmatic environment for RDF, RDFS and OWL, SPARQL and includes a rule-based inference engine [5]. Jena is open source and grown out of work with the HP Labs Semantic Web Programme.

3 Construction of Pomology Domain Ontology

At present, ontology construction methodologies have not been standardized, there are numerous frequently quoted approaches. In this paper, thesaurus-based approach for building domain ontology [6] is improved, an approach to build PDO based on Agriculture Science Thesaurus (AST) is proposed. The process of it consists of the following phases:

3.1 Confirming Core Concepts

Thesaurus is made up of terms and the relationships between terms of certain domain, so the terms of domain ontology can be selected from it. In this case, core concepts of pomology domain are collected according to Agriculture Science Thesaurus (AST). Firstly, “Fruiter Crop” is selected as the first concept, and then “Fruiter Crop” is classified into eight genres: such as “Kernel Fruits”, “Berry” and so on. These terms are all the sub-concepts of “Fruiter Crop”. Secondly, the concepts relating to “Fruiter Crop” are selected. Table 1 gives the names and explanations of these concepts.

3.2 Adding the Properties of Concepts

Every confirmed core concept has many different properties to be added. For example, “name”, “address”, “telephone” and so on are added as the properties of “Academic Institution”. In addition, properties of concepts can be inherited by their sub-concepts.

3.3 Confirming the Relationships between Concepts

By semantic analysis, the relationships between concepts can be classified into two genres:

1）Hierarchical relationships

According to part/whole relationship from AST, hierarchical relationships of concepts including broader/ narrower and instances relationships are confirmed.

2）Nonhierarchical relationships

For example, Table 2 gives the nonhierarchical relationships between "Expert-Scholar" and other concepts.

Table 1. The general concepts of pomology domain

Name	Explanation
Fruiter Crop	Refers to the concepts of fruiter species
Expert - Scholar	Reflects the personal and academic information of researchers who study pomology in china.
Academic_ Institution	Reflects information about academic institutes, research institutes, and associations related to pomology.
Establishment	Instances :sprinkler, equipment, greenhouse and so on.
Environment	Soil, landform , sunlight .etc
Research project	Reflects the projects that are worked by Expert-Scholars
Fruit Breeding	Reflects skills of fruit breeding
Fruit Planting	Reflects means and conditions of planting

Table 2. The nonhierarchical relationships between "Expert -Scholar" and other concepts

Name	Explanation	Opposite
Department	Reflects relationship between "Expert-Scholar" and "Academic Institution"	Researcher
Implement	Reflects relationship between "Expert-Scholar" and "Fruit Breeding"	Executant
Research Direction	Reflects relationship of "expert - scholar" studying on "Planting" ," Breeding" and "Fruiter Crop"	Be studied
Using	Reflects relationship between "Expert - Scholar" and "Establishment"	Be used
Choosing	Reflects relationship between "Expert - Scholar" and "Environment"	Be chosen

3.4 Adding the Instances of Concepts

It is necessary for building domain ontology to supply the instances of concepts. For example: in this case, "Shandong Agriculture University", "Fruiter Association" are added as the instances of "Academic Institution". Further, the values of all properties of instances are supplied.

3.5 Representing Pomology Domain Ontology

In this case, protégé is selected as the developing tool, so the PDO can be represented by OWL. Further illustrate below:

Firstly, according to the confirmed core concepts, corresponding classes of PDO are created by using protégé. Classes hierarchy structure of PDO can be shown by OWLVizTab in protégé. Secondly, data properties and object properties of classes are added. Nonhierarchical relationships between concepts can be represented by means of adding object properties. For example: Fig.1 represents the relationships between "Expert –Scholar" and other classes by JambalayaTab. Finally, instances of classes are added. Since relationships can be inherited, instances have the same relationships (see Fig.2). After that, an owl file (Pomology.owl) is created by protégé.

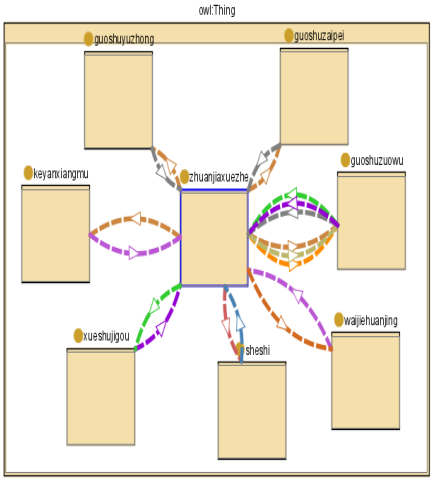

Fig.1. Non-hierarchical relationships between Expert-Scholar and other classes

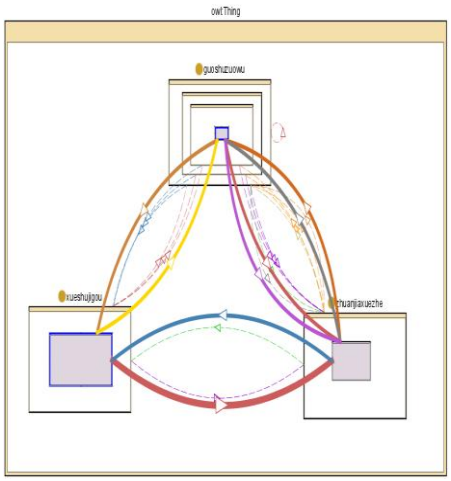

Fig.2. Non-hierarchical relationships between three instances

4 Design and Realization of Query System Based on Pomology Domain Ontology

4.1 Architecture of Query Model

In order to design query system, the model of it is built at first .The model consists of five modules, which are shown in the Fig.3. Further illustrate below.

By using interactive query interface, users can customize query conditions, which are sent into query processor. The functions of the query processor are executing corresponding algorithm and calling Jena methods according to the given query conditions, further, this processor supports three kinds of query, which are class

query , property query, and instance query. Parsing ontology file, it is a way to access and draw information from ontology file. After reading ontology file, the information of ontology classes, properties, and instances can be obtained and saved into storage structure by calling the Jena methods. Finally, the query results can be outputted in the visual interface.

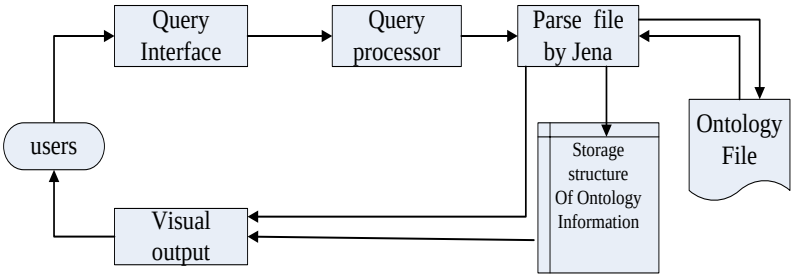

Fig.3. Flow-process diagram of query model

4.2 Parsing the Pomology Domain Ontology

Jena is used to parse the PDO in this study. The phases are as follows: firstly, an ontology model is created through the Jena ModelFactory.createOntologyModel(); secondly, using the read () method, an ontology document (Pomology.owl) is loaded into the created ontology model; finally, the owl file is parsed by using Jena API.

The methods of Jena that are used in this study are listed below:

All direct subclasses of ontology class can be found by calling listSubClasses (). In order to represent PDO hierarchy as tree structure, an algorithm is designed. The details of it are as below: “owl-thing” is set as root-node of the tree, and then set direct subclasses as child nodes of the root-node, in the end, all subclasses of PDO are obtained through using recursive algorithm. In addition, by using listIndividuals() method, all instances of the PDO can be returned; and all properties can be returned by the method listAllOntProperties().

4.3 Realization of Query System

This query system realizes three kinds of query, which are class query, instance query, and property query. Further illustrate below:

1) Class query

Firstly, class query realizes functions of class hierarchy queries. Methods of query are as follows: given an ontology class object, a list subclasses of this class are obtained by calling listSubClasses(); the direct superclass of this class is obtained by calling getSuperClass(); and a list instances of this class are obtained by calling listInstances(). In addition, class query lists properties of certain class by using listDeclaredProperties().

Secondly, class query supports interactive query [7], which can execute query on classes according to several conditions demanded by users. Further illustrate below: given the name of property, a list of classes related to this property are saved as to a arraylist by calling listDomain(); then a subclasses collection of the given superclass are saved as to another arraylist by calling listSubClasses(); Finally, the intersection of two arraylists is obtained, which are classes that have given superclass and property together.

The class hierarchy tree is on the left side of visual query interface, and the interactive interface is on the right side. While user clicks a random treenode of this tree, subclasses, superclass, properties and instances of this class are displayed on the right side (see Fig.4). Further more, by clicking "query" button, users can make use of interactive interface, in which several query conditions can be selected from JComboBoxs, and results of query are displayed.

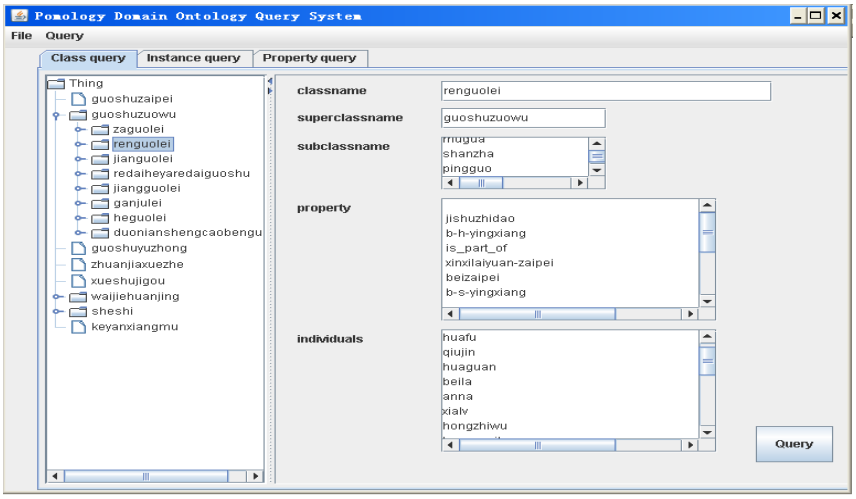

Fig. 4. Class query interface

2) Instance query

Firstly, the function of finding out all properties of the given instance is realized, details of the method are as follows: given instance, the listProperties() method can return a set of statements (triples), the predicate of a statement (property) can be got by using statement getPredicate() method , the Object of a statement can be obtained by calling getObject()[8],the algorithm is as follows:

```
NS=omx.getNsPrefixURI("");//omx is ontology model
note1=(DefaultMutableTreeNode)
jTree2.getLastSelectedPathComponent();
Individual in=omx.getIndividual(NS+note1);
for(StmtIterator ip=in.listProperties();ip.hasNext();)
{ Statement out=(Statement) ip.next();
  String p=out.getPredicate().getLocalName().toString();
```

```
OntProperty a1=omx.getOntProperty(NS+p);
s=s+p;//String s
RDFNode tt= out.getObject();}
```

Since the Object of a statement can be either a resource or literal, the method returns an object typed as RDFNode, the algorithm of processing RDFNode is as follows:

```
if (tt instanceof Resource) //judge RDFNode tt is
Resource
{ s=s+"  value "+(((Resource)
tt).getLocalName().toString()); }
else { s=s+" Value \"" + tt.toString() + "\""; // tt is
a literal}
```

Secondly, interactive instance query supports query on instance according to several conditions demanded by users. Given ontology class, several names and values of properties, all instances of given class can be found and saved to a arraylist, and then instances that have given certain properties and values can be queried too, the query results of different properties are saved to different arraylists, finally, the intersection of all arraylists is obtained.

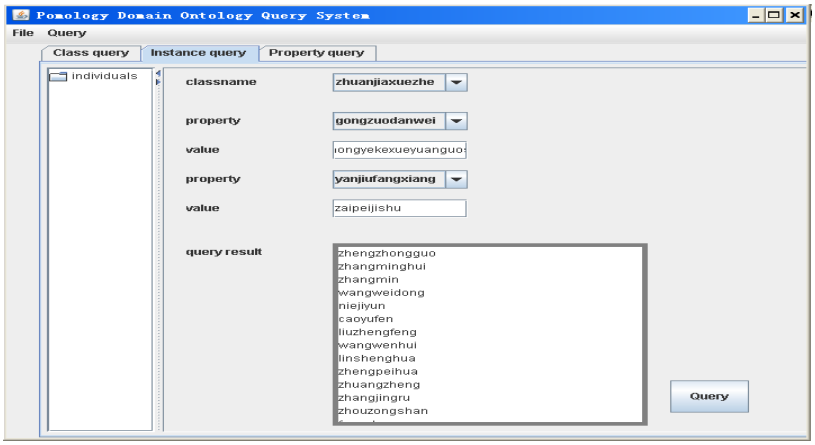

Fig.5. Instance query interface

The list instances of the PDO are displayed on the left side of query interface, while user select one from the list, all properties of the certain instance are displayed. By using the interactive interface, users can accomplish interactive query. For example, while user input "zhuanjiaxuezhe", "gongzuodanwei" 'zhongguonongyekexueyuanguoshusuo", "yanjiufangxiang", "zaipeijishu" in turn, the query results are expert-scholars who research on fruit planting and work in Pomology Institute of Chinese Academy of Agricultural Sciences (see Fig.5).

3) Property query

The property query realizes returning domain of the given property by calling getdomain(), and the range of the property by calling getrange(). Further, this query supports query on property according to given domain, range, type and so on.

The list Properties of the PDO are displayed on the left side of query interface, while user select one from the list, the domain and the range of the certain property are displayed.

5 Conclusion

In this paper, the modeling of PDO is studied, design and realization techniques of the query system based on PDO are proposed. This query system realizes class query, property query, and instance query by using Jena, further more, it supports interactive query. Through it pomology knowledge can be obtained from PDO according to user needs. Query results show that the query system based on PDO is practical for users to query information from PDO.

Acknowledgements

I would like to express my gratitude to all those who have helped me during the writing of this thesis. I acknowledge the help of Professor Liang Yong. I do appreciate his professional instructions. I would like to thank Wu Qiulan, who kindly gave me a hand when I was meeting difficulties.

Last but not the least, my gratitude also extends to my family who have been assisting, supporting and caring for me all of my life.

References

1. Li, H. Q.: Ontology Storage and Querying Technology. D. Beijing University of Posts and Telecommunications (2007)
2. Resource Description Framework, http://www.w3.org/RDF/
3. Bechhofer, S., Harmelen, F.V., Hendler, J.,et al.: OWL Web Ontology Language Reference. W3C Recommendation 10 February 2004. http://www.w3.org/TR/owl-ref/
4. Protégé, http://protege.stanford.edu/
5. Jena A Semantic Web Framework for Java, http://jena.sourceforge.net/
6. Tang, A.M., Zhen, Q.: The Study on Thesaurus-based Construction of Domain Ontology. J. New Technology of Library and Information Service. 41-45 (2005)
7. Wu, J. L.: Research and Implementation of Semantic Retrieval System based on Domain Ontology. D. Taiyuan University of Technology (2010)
8. Sheng, Q.Y., Yin, G.S.: Jena-based Dynamic Semantic Retrieval Method. J. Computer Engineering. Vol.35, No.16, 62-64 (2009)