

A Method for Information Management Based on RDF Model and Ontology Technology

Daoqu Geng, Guan Gong*, Qi Gao, Yi Wen and Chuntang Chen

College of Automation, Chongqing University of Posts and Telecommunications, Chongqing, China

*Corresponding author

Abstract—The approach of Internet of Everything brings some challenges to the storage and processing of those connected data. Aiming at problems of complex construction process, tedious maintenance of relational table and weak expansibility of table structure when using traditional relational database to store the data, we propose an information management method based on RDF data model and ontology technology. Firstly, the data model used in this method can simplify modeling and operating of the data, it also has flexible ability for property expansion. Secondly, combined with ontology reasoning technology, this method ensures the consistency of information model, and provides implicit meaning of data for information service.

Keywords—RDF; data modeling; information management; ontology technology

I. INTRODUCTION

As early as 2001, Berners-Lee proposed the concept of Semantic Web [1], advocating the expression of data on the Internet in a machine understandable way. At the same time, data were linked to build a web of data with large scale and rich links. As the physical world gradually entered the era of Internet Of Everything [2], and the increasing number of interconnected objects, the links among them become complex and varied. However, Lacking of a simple and efficient way to storage and process these internet data occurs nowadays. Traditional relational databases often need to create a series of associated tables to record the relationships among these data when they need to describe their connections. This will lead to low storage efficiency and difficult data maintenance [3], [4]. In recent years, along with the development of social networking, NoSQL database technologies have gradually matured. Databases using those technologies are various, such as Redis, DynamoDB, etc. of the key-value storage model; Apache HBase, Cassandra, etc. of column storage model; MongoDB, CouchDB, etc. of document storage model. But there exist problems of query efficiency and reasoning support when these databases are used to store RDF data [5].

The data model based on graph structure can effectively solve the above problems. At present, many open or closed graph databases have been developed, such as Neo4J [6], AllegroGraph [7], etc. Based on this graph model and ontology technology, we propose a data information management method, which is able to operate data conveniently and improve the intelligence level of data information management. This paper firstly introduces related knowledge, and then gives the description of information management structure based on graph model and ontology, and introduces the main technology in this

structure. Finally, the storage management of graph database and the application of ontology reasoning are tested and analyzed. And last part is the conclusion.

II. BACKGROUND KNOWLEDGE

A. Resource Description Framework

RDF(Resource Description Framework)is a data model for describing resource and it is based on graph structure. In this framework, data are described as triple: subject, predicate (property) and object (value). A series of those triples form a RDF graph, and the nodes in RDF graph can be a URI reference, literal, or empty node. RDF data that follows this framework can be formatted as text form of N-Triples, Turtle and RDF/XML stored in the local file system, and can also be directly imported into graph database when needed. As an example of a *Cup* information, a RDF data model is constructed as shown in Figure 1, and the composition of a cup and its related properties-values are stated.

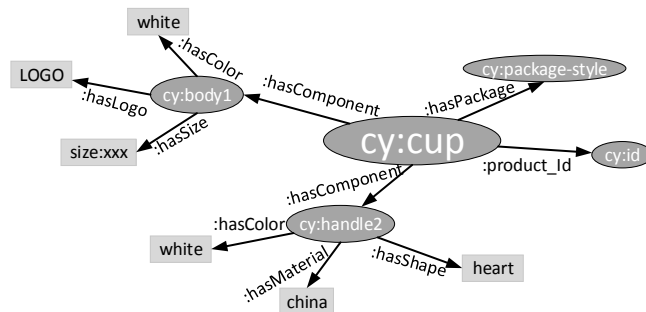


FIGURE 1. OBJECT INFORMATION DESCRIBED BY RDF TRIPLES

B. Prolog

Prolog (Programming in Logic) is a logical programming language based on first-order predicate. Unlike other programming languages, Prolog is used as a declarative programming language and express facts and relations. Computation of Prolog program is initiated by running a query over these relations and it can be integrated with OWL ontology [8]. For example, when following facts are declared in Prolog:

```
product(cup).  
component(body 1).  
hasComponent(cup, body 1).
```

It means that cup is a product, while body 1 is a component, and Cup has a component body 1. Specially, "<--" is used to connect conditions and conclusion within AGProlog syntax [7].

C. Ontology

The concept of ontology is derived from the philosophical field, which is generally accepted in information science as a clear formal description of the shared conceptual model [9]. Ontology is the core of semantic web technology, it has strong ability of logical description and knowledge expression. Ontology enhances the ability of knowledge expression by defining the concepts and terms in a certain field and by combining these terms and relationships to define the rules of lexical epitaxy. OWL is a language used to edit ontology, which is standardized and recommended by the W3C. Currently, visual ontology construction tools that support this language are Protégé [10] and Top Braid Composer [11], etc.

III. GENERAL ARCHITECTURE

An overall structure of the information management based on the graph data model and the ontology technology is shown in Figure 2.

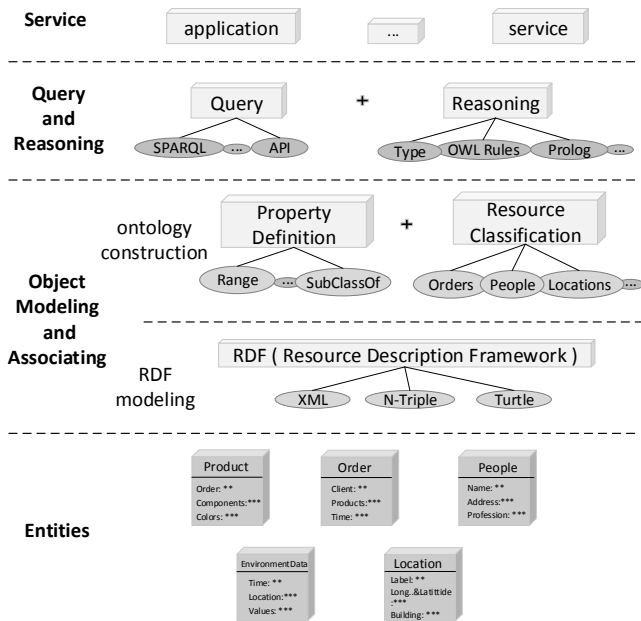


FIGURE II. STRUCTURE OF THE INFORMATION MANAGEMENT METHOD BASED ON RDF MODEL AND ONTOLOGY

The first layer includes entities in a specific application. These entities are producers of data, not only logical entities, such as orders, but also physical entities, such as products, people, etc.

The second layer is a process of object modeling and association. Entities above and their relationships are formalized in information world through certain knowledge expression, and become the mapping of entity objects in information world, reflecting their state of entities.

In the third layer, Query and reasoning are performed. After abstraction of data information model in the second level, this layer can implement query and access to objects, and also acquire hidden knowledge and information under rule reasoning or ontology reasoning.

The fourth layer is a service layer. Based on the data information obtained directly or reasoned out, it provides specific service for data consumers.

IV. TECHNICAL STRUCTURE

A. Data Modeling

Data modeling includes two aspects: object modeling and ontology construction. Object modeling is to model entity information that needs to be managed, while ontology construction is a process of classifying resources using domain knowledge and constraining on the relations between those classes through defining properties.

1) *Object modeling*: Each entity processed in the information world has fixed or dynamic properties and values, and object Modeling needs to describe those properties of entities. The method for object modeling based on RDF accords with the characteristic structure of entity itself, and brings flexible expansion ability for new properties. For example, in information management of products, a *cup* is known to have order number *o:1699*, and the user to which the order belongs is *u:guan*, as shown in Figure 3. If *belongsTo* is defined as a transitivity property in the application ontology, a new fact that the cup *belongsTo* user *u: guan* is got.

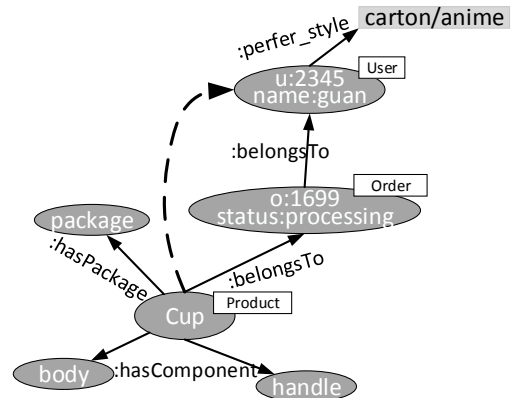


FIGURE III. EXTENSION OF BELONGSTO PROPERTY

Then, a relationship between the product and the user can be established. Moreover, a similar style of packaging can be developed for a product in accordance with *u:guan*'s preferred style. Similarly, when all components of a product are declared as facts, the following Prolog rules (AGProlog rules):

```
<--(hasColor ?product ?color).
    (?product !:hasComponent ?Component).
    (?Component !:hasColor ?color).
```

can be run to get the color information of this product, and a better information support for product recommendation services can be provided.

2) *Ontology construction*: As the concept of ontology is introduced in section 2, an ontology construction method using OWL language is introduced here. This method is based on the seven-step constructing process[12] from the Stanford University: Determine the domain and scope of the ontology,

Consider reusing existing ontologies, Enumerate important terms in the ontology, Define the classes and the class hierarchy, Define the properties for classes, Define the axioms of constraint for classes, Create instances and Check the consistency. Based on the process above, ontology construction can be completed in an editor, while constraints axioms described by OWL language need to be tested and checked by a reasoner. For example, when building a constraint axiom between product and order, an order can belong to only one user. Following Manchester syntax in Protégé, the order is defined as fellow:

$:Order \equiv \exists :belongsTo \max 1 :Person.$

It means that, the maximum number of person to whom a product belongs is 1, and a consistent detection error will occur once the fact that the owner of an order is not unique is reasoned. Other quantitative constraints can be defined using constraint words such as some, min, only, exactly.

B. Storage and Query

The storage of RDF data used in this article is based on graph model. Graph model can directly model RDF data and provide efficient query and reasoning support on those data. To query the data, SPARQL language has been defined by the W3C, and there are four kinds of SPARQL query: CONSTRUCT, DESCRIBE, ASK, and the most commonly used SELECT. When using graph models to manage data, there are different management ways. In this paper, the way using default graph and the one using named graphs are compared and analyzed on their query time, and results is shown in the first part of section 5.

C. Ontology Reasoning

Ontology reasoning has two basic functions: One is to check the consistency of an ontology, which will detect whether a class or instance has logical consistency, conflict between types, etc. As mentioned in the order example, once an instance of Order: belongs to two objects of Person, an OWL. Same As relation between the two persons will be reasoned, otherwise there will be an ontology conflict and a further process is need. The other function is based on axioms defined in ontology, and is to draw some hidden facts under the existing ones by running an ontology reasoner. To illustrate an application of axiom reasoning, the color information, an extended property of the product, is used here. When Prolog rules for property extension in section 4.1, the color information of 7 preconfigured products is obtained as showed in Table 1.

TABLE I. COLOR INFORMATION OF PRODUCTS AFTER EXTENSION

Product	Values of <i>hasColor</i> property
Product_1	white
Product_2	white & black
Product_3	black & green
Product_4	white & green & gray
Product_5	black & green & gray & yellow
Product_6	white & black & green & gray & yellow
Product_7	blue

To meet a special need of color, an ontology about products color is constructed including axioms below:

$Product_1 \equiv \exists hascolor \text{ some white};$

$Product_2 \equiv \exists (hascolor \text{ some white}) \text{ and } (\exists hascolor \text{ some black});$

$Product_3 \equiv \dots \text{ (like above)}$

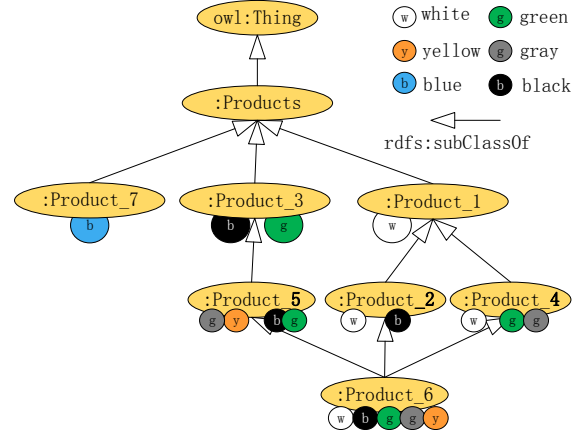


FIGURE IV. COLOR STRUCTURE OF PRODUCTS AFTER ONTOLOGY REASONING

After reasoning on the ontology of products is executed, a graph structure about color information of products will be obtained as shown in Figure 4. Based on this graph structure, a color-based analysis of competencies can be conducted to provide product recommendations for specific need of color. The second part of section 5 defines two different needs and gets the corresponding result.

V. TEST AND ANALYSIS

A. Storage Test

The software environment for testing the two management of graph data mentioned in section 4.3 is Java command line. Graph database uses the Allegro Graph database version 6.1.2 running on CentOS 6.9 operating system. The hardware environment includes: Shuguang A840-G10 server, AMD Opteron 6344 processor, 504G available memory, and 2T disk space. In the test, two kinds of data management on the same scale were queried by a SPARQL clause asking the same property values. The two ways are alternately carried out and circulated for 20 times, and the average query time of each of them is counted. Under the same query results of two ways, statistics of query time is shown in Figure 5. Abscissa indicates the size of queried RDF database and ordinate is the query time (MS). As we can see, because the way using named graph reduces the search space, it returns query results in a shorter time than the way using default graph.

B. Reasoning Result

Ontology reasoning requires the support of a reasoned which can be used or integrated according to the needs in specific applications. The Hermit reasoner used in this part is an efficient reasoning machine built in Protégé5.1 and it is implemented under the open world assumption.

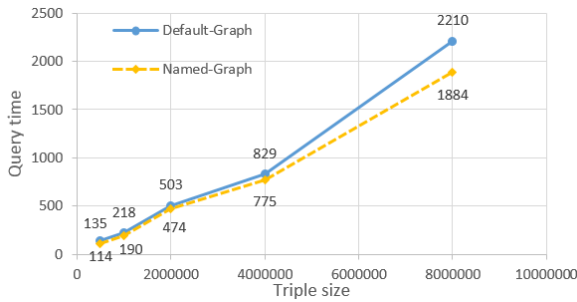


FIGURE V. QUERY TEST RESULTS OF GRAPH DATA MANAGEMENT

The graph structure of color information obtained in section 4.3 can be used to recommend products that meet the color requirements of user by building an intermediate class in the ontology. For example, when a need of product requires a white color property, an intermediate class can be constructed as follows to meet the need:

$Product_with_white \equiv \exists hasColor\ some\ white.$

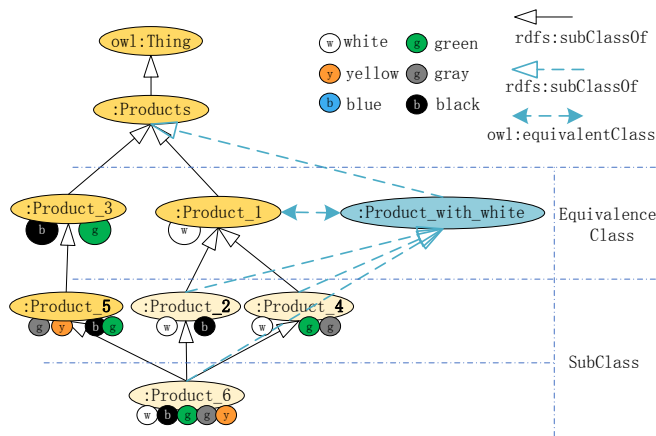


FIGURE VI. RESULTS OF ONTOLOGY REASONING ON THE INTERMEDIATE CLASS

Repeat a reasoning on the ontology and result is shown in Figure 6. Among the result, there is one equivalence class (*Product_1*), two direct subclasses (*Product_2* and *Product_4*), and one grandchild class (*Product_6*) of the *Product_with_white*. This parent-child class structure of products can be recommended layer by layer and meet the need from user.

VI. CONCLUSION

This paper introduces an information management method based on RDF graph model and ontology technology, which takes advantage of the graph structure in terms of model construction and property extension to manage connected and changeable information. The RDF data is used to model entities in the method, and improve extensibility of property. The proposed method combines ontology techniques to obtain richer data meaning and meet specific needs. Meanwhile, the two management ways of graph data are compared, and the way using Named Graph is recommended firstly as it reduces the search space.

REFERENCES

- [1] T. Berners-Lee, J. Hendler, and O. Lassila, "The Semantic Web," Scientific Am., May, 2001, pp. 34–43
- [2] R. Chandhok, The internet of everything, 2014 IEEE Hot Chips 26 Symposium (HCS), Cupertino, CA, USA, 2014, pp. 1-29.
- [3] SHEN Jian-tong. The Expression of Resource Description Framework (RDF) Data in RelationalDatabase [J]. SCI-TECH INFORMATION DEVELOPMENT & ECONOMY, 2008, 18(25):139-140.
- [4] YI Ya-xin ,SONGZi-lin ,YIN Kang-yin. Research and Implementation on Storage Schema of RDF Data [J]. Information Science, 2007, 25(8):1218-1222.
- [5] WANG Lin-bin, LI Jian-hu, SHEN Zhi-hong. Overview of NoSQL databases for large scaled RDF data management [J]. Application Research of Computers, 2015(5):1281-1286.
- [6] Neo4J [EB/OL]. <http://neo4j.com>.
- [7] AllegroGraph [EB/OL]. <https://franz.com/agraph/allegrograph>
- [8] Vassiliadis V, Wielemaker J, Mungall C. Processing OWL2 ontologies using thea: an application of logic programming[C]// International Conference on Owl: Experiences and Directions. CEUR-WS.org, 2009:89-98
- [9] Studer R, Benjamins V R, Fensel D. Knowledge engineering: principles and methods. Data Knowl Eng 25(1-2):161-197[J]. Data & Knowledge Engineering, 1998, 25(1-2):161-197
- [10] Protégé [EB/OL]. <https://protege.stanford.edu/>
- [11] Top Braid Composer [EB/OL]. <https://www.topquadrant.com/tools/modeling-topbraid-composer-standard-edition/>
- [12] Natalya F. Noy and Deborah L. McGuinness. Ontology Development 101: A Guide to Creating Your First Ontology. [EB/OL] https://protege.stanford.edu/publications/ontology_development/ontology101-noy-mcguinness.html.