Research on the Development Method of the Semantic Web Application Based on AllegroGraph

Daoqu Geng and Yi Wen

ABSTRACT

With the continuous development of semantic web technology, semantic web applications are getting more and more attention. However, the research on semantic web applications at home and abroad is still in the theoretical stage, lacking the specific methods of semantic web application development. Moreover, there are few semantic web products for users. In response to this problem, the advantages of AllegroGraph in semantic data storage and query as well as in the semantic web application development are introduced. Then the development framework of the semantic web application based on AllegroGraph and its implementation method are given. Specifically, taking information retrieval as an example, the construction process of the data layer, the logic layer and the presentation layer are described in detail. The experimental results show that the information retrieval method based on semantic, compared with that based on keyword, has a significant improvement in the recall rate and precision rate, which lays a foundation for the application of semantic web products in enterprises to a certain extent.

KEYWORDS

AllegroGraph, Semantic web application, Development method, Semantic information retrieval.

INTRODUCTION

With the continuous development of semantic web technology[1], semantic web applications have emerged. These applications take advantages of the semantic web to highlight the “data-centric” programming philosophy, enabling the information on the network to have computer-understandable semantics, thus realizing information sharing and reuse, and supporting abundant information operations. At present, the research on semantic web applications has shifted from the theoretical stage to the practical application level. For example, the database supports the storage and query of RDF (Resource Description Framework) and OWL (Web Ontology Language) documents. The browser starts to support the analysis of RDF and other ontology documents. Therefore, more and more attention has been paid to the research on semantic Web applications.

At present, the research on semantic web applications at home and abroad is in the theoretical stage, lacking the specific methods of semantic web application development.
development. Moreover, there are few semantic web products for users. Therefore, it is of great significance to study the development methods of semantic Web applications. Yan D-W et al. proposed a semantic retrieval method based on ontology, which uses Jena framework to develop semantic web applications[2]. However, this method mainly focuses on the description, construction and storage of ontology, and lacks the research on the development method of the semantic web application. Sun X-J et al. gives the definition, composition and nature of the semantic web application, and establishes the development model of the semantic web application[3]. Nevertheless, the development model only gives the abstract development framework, lacking the specific development method of the semantic web application. Li X-L et al. gives the definition, structure and development method of the semantic web application, and illustrates the structural characteristics and construction process of the semantic web application based on the three-tier architecture[4]. However, the development method has certain limitations, and other applications written in programming languages lack compatibility. Therefore, this paper presents a framework of semantic web application and its implementation method based on AllegroGraph. The framework is compatible with the popular Jena, Sesame and other excellent programming frameworks, which lays a foundation for the subsequent application of semantic web products in enterprises. The AllegroGraph is not only a database technology, but also a framework for building semantic web applications, which supports a variety of client and programming languages based on REST communication protocols, including Java Sesame/Jena, Python, Lisp and C#. Unlike traditional relational databases or NoSQL databases, AllegroGraph combines semantic and graphical techniques to intelligently process data through context and concepts to maximize information utilization.

SEMANTIC WEB APPLICATION DEVELOPMENT BASED ON ALLEGROGRAPH

DEVELOPMENT ARCHITECTURE

The development architecture of the semantic web application based on AllegroGraph is shown in Figure 1, which provides a semantic web database for storing RDF data, and uses various APIs supported by AllegroGraph to provide a consistent, complete and flexible programming environment for semantic web applications. The development architecture transforms classes, properties and instances in a semantic web application into classes, objects and methods used in object-oriented programming. The development architecture adopts a three-tier architecture mode of existing web applications[5]: a data layer, a logic layer and a presentation layer. The data layer includes an ontology knowledge base and a rule base. The logic layer includes a semantic query module and a semantic inference module, and the presentation layer includes a user interface.
The AllegroGraph-based Semantic Web application development framework differs from the existing Jena/Sesame programming framework in the following areas:

1) Database. The relational database query used by the existing Jena/Sesame programming framework is less efficient, and the semantic index is more expensive. The development framework presented in this paper uses AllegroGraph as a semantic web database. It is a modern, high-performance, persistent semantic graph database that combines semantic and graphical techniques to intelligently process data through context and concepts to maximize information. The AllegroGraph database can store billions of triples/quads while maintaining excellent performance.

2) Application interface. The application interface of the existing Jena/Sesame programming framework is single, which hinders the development of diversified semantic web products. However, the development framework presented in this article is based on the REST communication protocol, which supports a variety of client application interfaces, such as Java Jena / Sesame, Python and Lisp. Moreover, it confirms to W3C / ISO standards and supports SPARQL 1.1, RDFS ++, SPIN, and Prolog rules/reasoning.

3) Semantic query. The development framework provides two semantic query mechanisms, SPARQL and Prolog. These query mechanisms can not only implement simple query of graph patterns, but also run complex queries, which can help users do some predictive analysis and make better and more real-time decisions.

4) Semantic reasoning. The development framework integrates a very fast and practical RDFS++ reasoner, which can be used for Prolog-based RDF reasoning.

SEMANTIC WEB APPLICATION INSTANCE DEVELOPMENT

Semantic information retrieval is one of the common semantic web applications[6]. In this part, the instance development of the semantic web application is introduced from three aspects: the data layer, the logical layer and the presentation layer.
DATA LAYER DEVELOPMENT

The development of the data layer mainly includes two parts: building an ontology knowledge base and creating inference rule base. The ontology knowledge base is used to provide data support for the semantic web applications. While the rule base is used to provide the basis for the semantic reasoning of the logic layer.

1) Building an ontology knowledge base.

The ontology building is the key step in the process of the ontology knowledge base construction, which adopts the “seven-step method” developed by Stanford University as the ontology modeling method[7]. Moreover, the TopBraid Composer (enterprise edition) is adopted to construct a literature-related ontology. The full-text database of CNKI is selected as the ontology instance data source, and 500 documents related to semantic web, semantic network, artificial intelligence and robotics are extracted and mapped into instance data. Part of the literature ontology model is shown in Figure 2. After the ontology is constructed, it will be stored in a database. This paper selects AllegroGraph as the storage ontology database, which can store and query ontology data in the form of RDF triples. In addition, the number of storage can be extended to billions, while maintaining excellent performance. This forms an ontology knowledge base with superior performance.

![Figure 2. Part of the literature ontology mode.](image)

2) Create a rule base.

The inference rule making is the key step in the process of rule base creation, which is based on the semantic relationships, hierarchical relationships and attributes among the concepts of specific knowledge ontology. Moreover, knowledge ontology has good conceptual hierarchy, information organization and knowledge expression ability. Moreover, the classes, attributes, functions, axioms and examples of knowledge ontology can express the relationships and constraints between the concepts in the described domain well [8]. Therefore, the knowledge ontology can be well combined with the established rules to meet the need of specific reasoning.

This paper uses the RDF Prolog provided by AllegroGraph as the rule description language, and the inference rules are expressed in the form of goal:-facts, where goal is the target and facts is the fact. For example, in the process of literature search, the problem of the same name author is often encountered. Moreover, the traditional keyword-based grammar information retrieval cannot give accurate results. Semantic-based information retrieval, however, can establish the relationship between them by formulating a series of
custom rules, and obtain more accurate and comprehensive results through semantic reasoning.

LOGICAL LAYER DEVELOPMENT

The logic layer development mainly includes two parts: the semantic query module and the semantic inference module. The semantic query module mainly accesses the ontology knowledge base and returns the result of the user request. The semantic reasoning module mainly infers the knowledge in the ontology knowledge base according to the inference rules in the rule base, and obtains the implicit knowledge.

1) Semantic query module.

The semantic query module is implemented by using the AG query mechanism provided by AllegroGraph. Figure 3 shows the framework of the semantic query module, which includes SPARQL and Prolog query mechanisms.

Figure 3. Semantic query module framework.

SPARQL is a standard RDF query language defined by the W3C that can be used to query different data sources. Typically, SPARQL constructs a query form by using the SELECT keyword and retrieves the expected data set from the database that stores the triples by graph pattern matching. In the basic graph mode, a simple query can be realized by defining the subject, predicate or object of each triple pattern as a variable to match the RDF triple set. Complex query can be realized by using the group graph pattern, the optional graph pattern, the multi-graph pattern and the named graph pattern.

Prolog query is another query mechanism provided by AllegroGraph. In fact, Prolog is a declarative programming language. Its program logic is represented in the form of relationships, and its calculations are performed by running queries against these relationships. The Prolog programming language based on formal logic and RDF triples describe the facts in a similar way. Moreover, Prolog supports customized rule matching and query, which makes it more convenient to extend knowledge beyond known information. This paper uses the Prolog query mechanism mainly for the semantic reasoning module, which is described in detail in the next section.

2) Semantic reasoning module.

Figure 4 shows the framework of the semantic reasoning module, which is divided into two parts: RDFS++ inference and custom rule inference. The semantic reasoning module can not only use the RDFS++ inference engine for logical reasoning, but also add custom rules in the rule base to reason beyond the logic deductive ability. Moreover, the semantic reasoning module is implemented by using the AGJena inference engine provided by AllegroGraph. AGJena is optimized based on the open source project Apache Jena to meet the needs of practical applications. When the user accesses the ontology knowledge base,
many hidden knowledge cannot be obtained. Therefore, the inference engine is required to reason to obtain the implicit knowledge.

![Diagram](image)

**Figure 4. Semantic reasoning module framework.**

**PRESENTATION LAYER DEVELOPMENT**

Presentation layer is mainly used to realize interactive access between users and semantic web applications. The key step is to design the user interface or browser page. Specifically, JSP and HTML are adopted to write browser pages for exchanging information between users and the knowledge base.

**EXPERIMENTAL VERIFICATION**

**EXPERIMENTAL ENVIRONMENT**

The experimental application development adopts Java language. The experiment is perform on Windows 7 64-bits PC equipped with an Intel(R) Core(TM) i5-4590 processor, 3.30GHz main frequency CPU, 8G memory, 500GB hard disk. The software used in the experiment and its functions are shown in the table I.

<table>
<thead>
<tr>
<th>Name of software</th>
<th>Functional description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Eclipse Jee Notes 4.8</td>
<td>Java web application development environment</td>
</tr>
<tr>
<td>Tomcat 9.0</td>
<td>Web application server</td>
</tr>
<tr>
<td>MySQL 5.7.18</td>
<td>Instance data storage carrier</td>
</tr>
<tr>
<td>TopBraid Composer</td>
<td>Ontology model building tool</td>
</tr>
<tr>
<td>AllegroGraph 6.1.2</td>
<td>Ontology storage carrier and development framework</td>
</tr>
</tbody>
</table>

**EXPERIMENTAL RESULTS AND ANALYSIS**

In the experiment, we compare the semantic information retrieval with the keyword-based grammar information retrieval. The precision rate and the recall rate are tested under different number of query condition. In order to minimize the error of the search results, the experiment are carried out multiple groups of similar retrieval and each group is randomly tested for 30 times. The test results of each group are the average value of the 30 measurements, and the average retrieval results of the multiple groups are taken as the experimental results. The test results of the precision rate and the recall rate are shown in Figure 5 and Figure 6 respectively.
It can be seen from Figure 5 that the precision rate of the semantic information retrieval is significantly higher than that of the keyword-based grammar information retrieval. Moreover, with the increase of the quantity of the retrieval condition, the precision rate of the semantic information retrieval increases continuously, but the precision rate of the keyword-based grammar information retrieval sharply decreases, and the gap becomes more and more evident. Specifically, the maximum difference reaches over 70%. Through analysis, we can see that there are two possible reasons for the above results. One is that the keyword-based grammar information retrieval has obtained a lot of information that is not directly related to the search content, which makes the precision rate lower. On the other hand, the matching degree between the results of the semantic information retrieval and query contents is higher.

Figure 6 shows that the recall rate of the semantic information retrieval is markedly higher than that of the keyword-based grammar information retrieval. Moreover, with the increase of the quantity of the retrieval condition, their recall rates are both increasing. However, the maximum difference is more than 40%. This is because the keyword-based grammar information retrieval is based on string matching, while the semantic information retrieval is based on semantic matching.

**SUMMARY**

This paper first introduces the advantages of AllegroGraph in building semantic web applications, then presents the framework and the development
method of semantic web application based on AllegroGraph, and finally elaborates the construction process of the data layer, the logic layer and the presentation layer. By constructing a semantic web application instance and comparing it with traditional web applications, the experimental results show that the semantic web application instance has a significant improvement in precision rate and recall rate. However, because semantic web application instances have semantic reasoning tasks, they take longer time to retrieve than traditional web applications. In the near future, we will focus on the efficiency of semantic reasoning.

ACKNOWLEDGEMENTS

This work is supported in part by the Project of Basic Research and Frontier Exploration of Chongqing Science and Technology Commission under Grant cstc2016cjyja0586, in part by the Project of Technological Innovation and Application Development of Chongqing Science and Technology Commission under Grant cstc2018jszx-cyzd0078.

REFERENCES