Two-step Role-Based Access Control method for Ontology Storage

Sukhoon Lee, Jangwon Kim, and Doo-Kwon Baik

1 Dept. of Computer Science and Radio Communications, Korea University, Seoul, Republic of Korea
leha82@korea.ac.kr, ikaros1223@korea.ac.kr, baikdk@korea.ac.kr

Abstract - Because of the advance of ontology technology and the explosion of web ontology, ontology repository has become a necessity. As the information security issue, research for adapting access control to the ontology storage has been studied, and there are two approaches which are model-driven approach and query rewriting approach. To solve the problems which these approaches have, this paper proposes two-step Role-Based Access Control method and describes a system architecture applying proposed method. The proposed method is efficient and reliable compared by the typical approaches.

Keywords: Access Control, Ontology Storage, Role-based Access Control

1 Introduction

the concept of Semantic Web was appeared by Tim Berners-Lee, ontology technology has been advanced to represent some knowledge or information [1]. Especially by W3C, the web ontology has been developed to represent ontology for web pages such as OWL/RDF [2, 3], and an amount of web ontology is increased explosively like web pages does. Due to processing and managing more ontology fast, the ontology storage which uses database system has been developed such as Jena [4], KAON [5].

Meanwhile, the web ontology has security issues like, is it validate that some information could access for a user? Because access control could not only restrict some systems, but also some information, it is suitable technology for the ontology storage of security issues that some information has to restrict to some users[6].

The researches for adapting access control techniques to the ontology storage have two approaches such as Model-driven approach and Query rewriting approach.

First, Model-driven approach executes SPARQL query in the ontology storage. As a result of query execution, ontology model is created. After inference of this model, the access control policies are adapted [7, 8]. This approach builds ontology model in memory as the result of query. It cause much cost and time because some ontologies are useless so that the ontologies are filtered in access control process. If a huge amount of ontology is loaded in memory as a query result and most of them are filtered in access control process, it takes much cost and time.

Query rewriting approach is a method of rewriting the SPARQL query before query execution in the ontology storage. The query rewrites to allow the information to user by authority [9-11]. This approach just rewrites query, so it is easy to adapt to any type of ontology storage which is able to SPARQL query. But after the query execution, the access control process is not supported about inferred ontology. Although this approach does not load useless ontology, it does not completely guarantee the privacy after ontology inference.

There are several the access control techniques. As one of them, Role-based Access Control (RBAC) divides users by roles, pairs objects with operations (access or deny) as permission, and define policies to pair of roles and permissions [12]. Because RBAC identifies users in web ontology and makes available access not only systems but also information, it is suitable technique to define policy for access control.

To solve pre-mentioned two problems, this paper proposes two-step role-based access control for ontology storage, implements and adapts to the ontology storage.

After this section, Section 2 describes two-step role-based access control method, and Section 3 describes the system architecture. Finally, Section 4 presents conclusion with evaluation.
2 Two-step Role-Based Access Control Method

The proposed method uses two-step access control process to deal with limitation of typical access control methods.

In the first step, it uses query rewriting method adapted policies by roles before query is executed in ontology storage. This step prevents loading the useless ontology in memory.

In the second step, it processes second access control after ontology model is expanded by inference. Without the second step, it does not guarantee the privacy about expanded ontology by inference, so the second step is necessary for reliability.

Figure 1 shows two-step Role-Based Access Control process as a proposed method.

2.1 Policy list creation

This process validates the policies which role of user has using user ID, and makes policy list consist of the policies which role of user has.

2.2 Query rewriting

This process rewrites given SPARQL query for adapting policy list which is made by previous process. As the first access control process, the inaccessible information from user is filtered in query level. So it is able to use resource efficiently because the useless ontology is filtered before model creation.

2.3 Query execution

This process creates ontology model by execution rewritten query. The ontology model is created in memory.

2.4 Model reasoning

This process expands the ontology model using inference. At this time, the inferred ontology may include inaccessible information from user.

2.5 Information restriction

This process does the second access control work using policy list which is made in first process. Because this process checks the policies about an expanded part of information, it guarantees the privacy about the inferred ontology.

2.6 Information representation

This process represents the ontology model which is formed by graph type in memory as ontology language like OWL/RDF due to supporting.

3 System Architecture

Figure 2 shows proposed system architecture. The proposed system consists of a proposed system and a database system. The proposed system inputs a SPARQL query from user, and communicates with database system to process the information.

3.1 Database System

Database system includes RBAC which works access control based on role and Ontology Storage which administer and store ontologies.
Figure 3 shows database model of RBAC. The policy defined in the RBAC consists of a pair of role and permission. Each user has a role and a permission defines whether some object is able to be accessed (operation) or not by user.

Figure 4 shows database model of Ontology Storage. The Ontology Storage is constructed to store triple statement. The triple statement consists of Subject, Predicate, and Object.

3.2 Proposed System

The proposed system consists of every modules in Figure 2 for input queries from user and two-step RBAC method.

- **User Interface module**: a module for user input/output. It receives user ID and SPARQL query from users, and return ontology in OWL/RDF type.
- **User management module**: a module for administration of user login and user information. It defines relation associated by User_Role in RBAC.
- **Policy definition module**: a module for defining a policy about a role. It defines relation associated by Permission in RBAC.
- **Policy list module**: a module for building policy list as searching policies which have role and permission pair by role of user.
- **Query rewriting module**: a module for rewriting SPARQL query. It rewrites query using policy list built in pre-process.
- **Ontology module creation module**: a module for executing query, creating ontology model, and reasoning the model. The ontology model loads in memory.
- **Information restriction module**: a module for the second access control of the ontology model using policy list in memory.
Information representation module: a module for representing ontology as ontology language like OWL/RDF to support ontology model to user.

4 Conclusions

This paper proposed and implemented the two-step RBAC method for overcoming the limitation of typical access control methods.

Table 1 shows a qualified evaluation of the two-step RBAC method compared with typical access control methods. The proposed method can faster and more efficient access control work than Model-driven access control method. And the proposed method have additional process as the second access control work for guarantee reliability about inferred ontology after query rewriting method.

<table>
<thead>
<tr>
<th></th>
<th>Model-driven</th>
<th>Query Rewriting</th>
<th>Two-step RBAC</th>
</tr>
</thead>
<tbody>
<tr>
<td>query performance</td>
<td>slow</td>
<td>fast</td>
<td>fast</td>
</tr>
<tr>
<td>resource usage</td>
<td>high</td>
<td>low</td>
<td>Low</td>
</tr>
<tr>
<td>reliability</td>
<td>reliable</td>
<td>unreliable</td>
<td>reliable</td>
</tr>
</tbody>
</table>

As the result, in an environment that administers and represents web ontology explosively increasing, the proposed method efficiently and reliably supports information when access control works.

In further works, we experiment useful dataset for implementing the proposed method, and determines performance and reliability for a qualified evaluation between the proposed method and the typical methods.

5 Acknowledgement

This study was supported by Second Brain Korea 21 Project and by the National IT Industry Promotion Agency (NIPA) under the program of Software Engineering Technologies Development. The corresponding author is Doo-Kwon Baik.

6 References


