Rule Management System for Ontology-based Recommendation System

Sarunyoo Ngamsrithepparit[1], Thepchai Supnithi[2], Ye Myat Thein[2],
Kanda Runapongsa Saikaew[1], Marut Buranarach[2], Seksan Poltree[1]
[1]Computer engineering, KhonKaen University, Thailand, 40002
[2]National Electronics and Computer Technology Center (NECTEC), Thailand, 12120
Email: sarunyoo@gmail.com, thepchai.supnithi@nectec.or.th, yemyatthein@gmail.com,
krunapon@kku.ac.th, marut.buranarach@nectec.or.th, seksan.poltree@gmail.com

Abstract— Rule management system was developed in order to provide a facility for domain experts to pass on their knowledge in rule-based format, by means of ontology concepts. Experts can utilize this system via web application, developed by deploying JSP and servlet technology. The web application contains the internal functions for dealing with ontology, database and rule file by deploying the JAVA language for which several APIs are available. One of JAVA APIs which the author employed is Jena API. Additionally, experts have ability to manage the rules such as writing new rule item, modifying the existing rules, removing the rules and so on. Author provides the tool that can assist experts to manage rules conveniently. For example, displaying the filtered results in the format of autocomplete, converting the filling data to make human-readable and rules repository can be done within rule file in the format of SPARQL syntax. Rules can be classified into four distinctive formats: messaging, instance insertion, instance creation, information search respectively. Besides, the web application has the capability of displaying the rule effect and information recommendation.

Keywords- Ontology, SPARQL, recommendation system and rule management system

I. INTRODUCTION

Nowadays, most search engine rests on keyword-based-search engine which stores the data in the database. The feature of database is non-structural storage device. Therefore, the efficiency of querying information is sensitive to vocabulary and low precision. In order to present the larger web content, the larger storage is required [9], particularly, the recommendation system [11] which requires the precise information to support the user decision. Therefore, Web ontology language (OWL) is required to use [5].This language is a part of Semantic Web [7] technology in order to implement this kind of recommendation system successfully. In fact, the individual ontology does not have the adequate capability of developing the intelligent agents [8].

In order to create the effective recommendation system, the main component that should be regarded for decision support is expertise knowledge of the given domain expert which are expressed in the rule-based format. One of the limitations of rule implementation is that the experts need to have the knowledge of querying language such as SPARQL [12] in order to manage the rule. The problem is that most of domain experts do not have sufficient knowledge in rule implementation. Therefore, rule-based programmer is required in order to express the knowledge representations in the format of rule. However, the programmer intervention needs time consumption, high cost and lacking of important knowledge in those domains. Therefore, Rule Management System for Ontology-based recommendation system is conducted.

The purpose of this recommendation system is to manage and aid the experts to share the knowledge with respect to the chosen domain without programmer intervention. The generated rules file enables the deployment with existing recommendation system. Rule management system also has the function to facilitate experts in real-time error checking.

The overview of this rule management system for ontology-based recommendation system is presented in the next Section II.

II. FOUNDATION OF THE RULE MANAGEMENT SYSTEM

The basic knowledge of this paper can be briefly summarized into five main categories as described in the following sections:

A. Semantic Web

The conventional web search engine seems to be suitable for human consumption due to the massive amount of human-readable contents. However, a user must spend time to search for the right content. Additionally, the proliferation of contents requires a time consumption effort for learner to access desired contents. Most of contents on the web are lacking the semantic parts. Therefore, it is difficult to find appropriate result when user requests. All of these problems can give utter dissatisfaction to the user, which any search service would want last. In order to meet the user requirements in querying the information, Semantic Web technology [7] offers the better way. It puts more emphasis on knowledge representation and management by adding the meaning or semantic into web contents in order to facilitate information search, extracting and representing the information vice and versa.

B. RDF

RDF [11] is used to describe the resources and identify the relationship between them. Resources are represented in the form of triple which are subjects, predicate, object respectively.

C. Ontology

According to Semantic Web technology, ontology uses the intelligent technique by representing the web content in the form of structural a.k.a machine-processable style, thereby implementing the concept of ontology [5] and as a result, improving the accuracy of web search. Theoretically, ontology defines a common vocabulary for researchers who need to share information in a domain which overcomes the differences in terminology, enabling reuse of domain knowledge to build a large ontology by integrating several existing ontologies to describe the portions of large domain.

D. Rule and inference

The ontology language can be viewed as specialization of predicate logic which can enhance the machine intelligence, tractable and is supported by efficient reasoning tool by implementing the Logic and interference or rule. According to the Logic, it is more general than ontologies since it can be used by intelligent agents for making decision and can provide more explanation than ontology language. Rule is liable for representing the relationships between facts. The inference system consists of inference rules of the form.

IF E contains certain triples

THEN add to E certain additional triples where E is an arbitrary set of RDF triples

E. Jena : Inference Engine

Basically, inference engine is a general program that activates knowledge in the knowledge base. In order to achieve the goal, author utilizes the Jena [6] which is the java framework. Jena is not only the framework that provides a programmatic environment for RDF, RDFS and OWL data, but also, provides the rule-based inference engine.

F. SPARQL

In order to access the meaningful data from RDF, the query language is required. In this paper, the SPARQL [12] language is chosen because SPARQL is the query language, recommended by W3C, which resembles like SQL constructs.

G. Recommendation System

Fundamentally, a recommendation system is the system that predicts user's preference to suggest items by the use of information filtering system technique. Rule management system for ontology-based recommendation [11] implemented the recommendation system to combine with rule management in order to enhance the system functional consistency. Moreover, this system enables to display the results retrieving each rule to experts for using the results in data validation immediately. Thus, implementing the rule to the recommendation system can diminish the errors which are performed from rule constructing.

III. Related Work

There are several researches which relate to recommendation system. The first recommendation system was developed in the midst of 1990s [13]. Many recommendation techniques have been used previously. Most of researchers employed collaborative filtering in their system in order to develop their recommendation system. The example of existing researches which relates to the recommendation system for example: Saman Shishehchi, Seyed Yashar Banihashem,Nor Azan Mat Zin presents a semantic recommendation system for e-learning domain in order to help the learners find subject [3] they need to learn by using ontology to present their learning contents. Their rule based system is based on OWL rules. They also employ the inference engine according to their rules.

The second finding is Ontology-based Association Rules Retrieval using Protégé Tools which were presented by Bin Shen, Min Yao, Zhaohui Wu, Yangu Zhang, and Wensheng Yi discussing ontology based association rule retrieval which can improve the current rule retrieval methods [10]. They also design and develop a prototype O-ARR for ontology based rules retrieval in ontology based rule base. Their O –ARR shows that ontology and Semantic Web can well-organize the rule knowledge.

Moreover, there are certain researches which exploites the concept of ontology and rule-based for user decision support. One of those researches is Decision-Making for Supplier Selection Based on Ontology and Rules which is presented by LuYiqing [4]. Their work on ontology based on SCOR and analysis rules of supply chain. Moreover, he also develops the coordination model based on ontology and rule for providing shared terminologies for representing general concepts and relationship of supply chains management domain.

Most researches tend to develop ontology along with rules. Nevertheless, there is still some trade-offs namely extending a research. Example of scenario: experts desire to insert rules; normally, most experts are absent of query language knowledge. Even programmer can implement the rule. However, he has insufficient knowledge of the domain as much as experts in specific domain. Thus, it is high likelihood for programmer to write incorrect rules. In order to solve these problems or some errors acquisition, this paper embraces the idea on using the rule management, which is beneficial for experts in specific domain who lacks of the query language in order to share their knowledge. This rule management system can allow the experts to create, edit, delete rule and make rule dependency. That is, it enhances the accuracy and reliability of knowledge based. In particular, it can help the experts minimize the error that may happen during developing the rules by utilizing the tool for displaying rule effect according to rules. Likewise, not only this system has ability of managing rules, but also, providing the recommendation system to facilitate the experts in creating rules for information recommendation or querying information within ontology.

IV. Construction of Rule Management System

In order to generate rule management system, there are several main architectural components, which can be classified into three parts: 1) user interface, 2) processing, 3) required data, which is in the RDF format respectively which can be illustrated in Figure 1. The user interface is that of the web-based application.

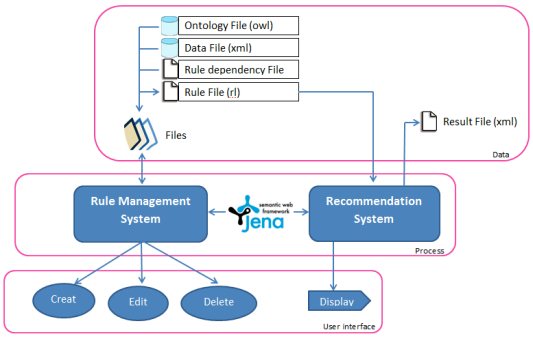

Figure 1: Rule Management System Architecture

The important portions depicted in Figure1 can be divided into smaller components following:

A. Design of Ontology

Sample ontology is the employee ontology in order to create the rule management system, inside ontology comprises of scope of knowledge base for employee and relationship of each class.

The Ontology was developed using Hozo software [1]. Hozo is a free software tool that was developed at Mizoguch Lab. and ENEGATE Japan for building and editing ontology.

The ontology was built by defining classes or subclasses and properties.

1) Classes or Subclass of ontology: Classes are defined using an owl: Class element. The class structure of the hozo ontology can be illustrated in Figure 2. For example, we can define a class Designing as follows:

```
<owl:Class rdf:ID="Designing">
   <rdfs:label>Designing</rdfs:label>
   <rdfs:subClassOf
   rdf:resource="#System_Development"/>
</owl:Class>
```

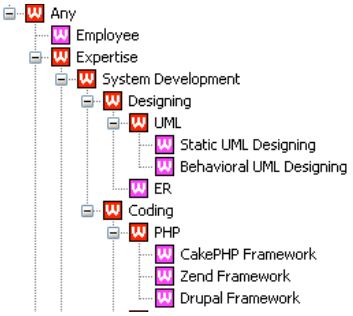

Figure 2: Class Structure Generated in Hozo

2) Properties of ontology: In OWL there are two kinds of properties

- Object properties, which relate objects to other objects.

  Here is an example of an Object property which is in the form of OWL:

```
<owl:ObjectProperty rdf:ID="has_Skill">
   <rdfs:subProperty Of rdf:resource="#hasPart" />
   <rdfs:domain rdf:resource="#Employee" />
</owl:ObjectProperty>
```

- Data type properties, which relate objects to data type values.

  Here is an example of a data type property which is in the form of OWL:

```
<owl:ObjectPropertyrdf:ID="has_eid">
   <rdfs:subPropertyOf rdf:resource="#hasAttribute"/>
   <rdfs:domain rdf:resource="#Employee" />
</owl:ObjectProperty>
```

Another example of properties in class Employee for knowledge base can be illustrated in Figure 3.

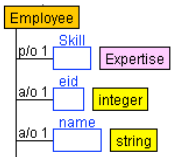

Figure 3: Properties in Class Employee

B. Database

This part is used as information storage. However, the ontology only stores the relationship of information. It does not have the capability of storing and editing the dynamic value. This paper utilizes the database to store data in the RDF format.

C. Rules

In order to create rules, basically, creating rule needs to rely on logic which has high flexibility. Thus, it is difficult to utilize the user interface in order to manage rules. Moreover, rules are generated by SPARQL language and are stored in the rule file. According to SPARQL language having the syntax in the form of query language, hence, it is quite challenging for an unskilled person in fetching and implementing this query language. We choose to implement the user interface based on essential cases according to the significant characteristic for usage to be several forms as following:

1) Messaging: Mainly, it defines the attribute by inserting the string to the attribute according to the defined condition. The example of messaging can be illustrated in Figure 4 and 5.

Rule name : employee.expert
If employee has skill is a PHP and employee has birthyear lessthan 1990 Then employee has expertise classification is “This employee is an expert in PHP.”

Figure 4: Statements in the Messaging Format

```
[employee.expert: (?x rdf:type ns:Employee) (?x
ns:has_Skill ?y0) (?y0 rdf:type ns:PHP) (?x
ns:has_birthyear ?y1) lessThan(?y1, 1990) -> (?x
ns:has_expertise_classification "This employee is
an expert in PHP.")]
```

Figure 5: SPARQL Syntax in the Messaging Format

2) Instance Insertion: It defines the property by inserting the existing data to the property. Alternatively, it may be said that interlinking the relationship according to the defined

conditions. The example of SPARQL in the Instance Insertion can be illustrated in Figure 6 and 7.

```
Rule name : employee.add.skill
If employee has eid equal 'larry.gg@xenoninc.com'
Then employee has additional skill is Expertise has
id equal 'kng_3'
```

Figure 6: Statements in the Instance Insertion

```
[employee.add.skill: (?a rdf:type ns:Expertise) (?a
ns:has_ExpertiseId ?b0) regex(?b0, 'kng_3')(?x
rdf:type ns:Employee) (?x ns:has_eid ?y0)
regex(?y0, 'larry.gg@xenoninc.com') -> (?x
ns:has_additional_skill ?a)]
```

Figure 7: SPARQL Syntax in the Instance Insertion

3) Instance Creation: This creates a new instance by creating a new cluster of information in order to prepare for applying. The example of SPARQL in the Instance Creation can be illustrated in Figure 8 and 9.

```
Rule instance_creation1
If Training_Program has_trainer_name EQUALS
'Dr.Error Gross' Then Training_Suggestion
has_program create new instance
instance_Training_Suggestion_1 That's
has_tsuggestion_name 'Dr.Gross Training'
```

Figure 8: Statements in the Instance Creation

```
[instance_creation1: (?x0 rdf:type ns:
Training_Program) (?x0 ns: has_trainer_name ?y0)
equal(?y0, 'Dr.Error Gross') -> (ns:
instance_Training_Suggestion_1 rdf:type ns:
Training_Suggestion) (ns: has_program ?x0) (ns:
instance_Training_Suggestion_1 ns:
has_tsuggestion_name 'Dr.Gross Training')]
```

Figure 9: SPARQL Syntax in the Instance Creation

4) Search: Relevance the rule for filtering or searching the knowledge-based in database which is in the form of rule-based that can be used for recommendation. The example of SPARQL in the search can be illustrated in Figure 10 and 11.

```
Rule name : employee.search.phone
If employee has skill is a PHP Then show employee
has phone number
```

Figure 10: Statements which is translated from SPARQL in the Search format in order to make human understandable.

```
[employee.search.phone: (?x rdf:type ns:Employee)
(?x ns:has_ Skill ?y0) (?y0 rdf:type ns:PHP) -> (?x
ns:has phone number ?z)]
```

Figure 11: The syntax of SPARQL language Search format.

A Not only a single rule base is required, but also, rules dependency Especially, in creating rules that represent instance creation that is unable to function independently, it is necessary to rest on the Instance Insertion rule-based in order to inserting the data from creating instance to the database which enables to manage by the use of rule dependency system.

D. Web Application Interface Designs

The user interface is web-based interface which is designed to be better than a conventional website. As an example, placing the data position can be done easily and customized to capture user requirements. Server-side processes run in local host by the use of Tomcat web container which supports JSP (Java Server Pages), the scripting language used for developing Java web application.

The background of web application interacting with ontology, the RDF based database and rule knowledge by the use of JAVA language in order to process the data. There are several ontology-processing APIs written in JAVA language. The paper chooses the Jena API for the implementation because it is widely used. Furthermore, we created a new API for specifically rule management in order to interact with web-based application.

The web application is designed so that it is easy to use and reduce the steps of rule management as well. The knowledge of ontology is not required for user. We employed the Ajax and jQuery technology to assist user for knowledge filtering according the domain of ontology into data field in order to minimize the error. According to rules implemented by using SPARQL language, we developed the autocomplete box to aid users for generating and manipulating the rule effortlessly. Moreover, this system enables to convert SPARQL rule to human-readable sentences. The example of the web application user interface is illustrated in Figure 12 and 13.

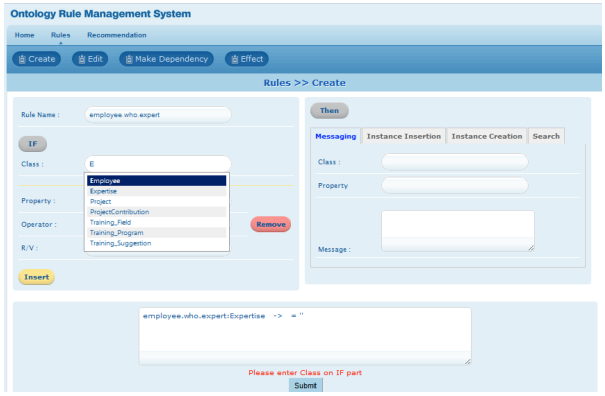

Figure 12: User Interface for Rule Creation

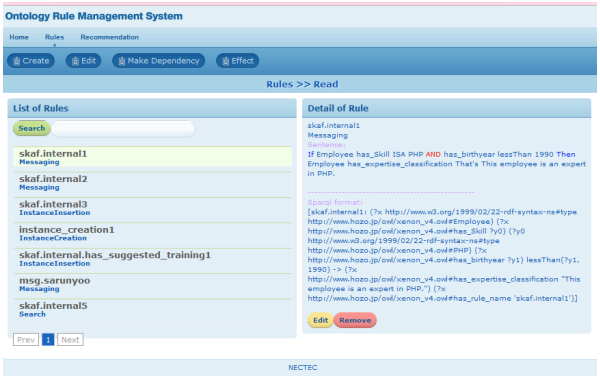

Figure 13: User Interface for Rule Parsing

V. CONSTRUCTION OF RECOMMENDATION SYSTEM

Recommendation system uses the implemented rules to analyze the raw data inside the database along with the knowledge in ontology in order to obtain the useful and precise knowledge.

Conventional recommendation systems comprise of the stand-alone system that can lead to take several steps in order to complete the tasks. In contrast, we developed the recommendation system to combine with the rule management system in order to make the system with better efficiency. As a result, it has the competency of inserting features to report the rule effect which is the main parts since it facilitates the experts to detect the correctness of rules immediately or to make them perceive the rule effect after implementing rule. All of these are the prominent feature.

The use of the recommendation can be achieved via web application. The functionality of this web application is fetching the rules to display in the form of search engine for using the specific recommendation.

VI. CONCLUSIONS

This paper proposes a rule management system for ontology-based in order to facilitate the experts in specific domain in sharing their knowledge in the rule-based format. The rules are managed (namely creating, editing, deleting and making rule dependency) by resting on Web application. This web application will function according to domain of ontology. Besides, the web application has the ability of displaying rule effect in order to facilitate experts in specific domain to enable to perceive the result of applying the rules. Additionally, the proposed system comprises of recommendation system used for querying information or information recommendation from database.

REFERENCES

[1] A.Felfernig, G. Friedrich and L. Schmidt-Thieme, "Recommender systems", IEEE Computer Society, 2007.

[2] Bin Shen, Min Yao, Zhaohui Wu, Yangu Zhang, and Wensheng Yi, "Ontology-based Association Rules Retrieval using Protégé Tools," Data Mining Workshops, 2006. ICDM Workshops 2006.,pp.765-769, 2006.

[3] Guo-chang Li, "Research on several Strategy based on Agent ofintelligent function in the application," Computer Science and Information Technology, 2009. ICCSIT 2009, pp. 542-545, 2009.

[4] J. Broekstra, A. Kampman, F. V. Harmelen, "Sesame: A Generic Architecture for Storing and Querying RDF and RDF Schema," Lecture Notes In Computer Science (LNCS), Vol.2342, pp. 54-68, Jun. 2002.

[5] Jeong-Dong Kim, Heeyoung Shin, Dongwon Jeong and Doo-Kwon Baik, "Jena Storage Plug-in Providing an Improved Query Processing Performance for Semantic Grid Computing Environment," Computational Science and Engineering Workshops, pp. 393-398, 2008.

[6] LuYiqing, Liu Lu, Li Chen, " Decision-Making for Supplier Selection Based on Ontology and Rules," Intelligent Computation Technology and Automation, 2009. ICICTA '09, Vol.4, pp. 176-179, 2009.

[7] M. Balabanovi and Y. Shoham, "Fab: Content-Based, Collaborative Recommendation," Communication of the ACM, vol. 40(3), 1997.

[8] M. L. Sbodioa, D. Martinb and C. Moulinc, "Discovering Semantic Web Services using SPARQL and Intelligent Agent," Journal Elsevier 2010.

[9] Mizoguchi Lab., the Institute of Scientific and Industrial Research, Osaka University, "Zoho-Ontology Editor", available from http://www.hozo.jp/

[10] M. Balabanovi and Y. Shoham, "Fab: Content-Based, Collaborative Recommendation," Communication of the ACM, vol. 40(3), 1997.

[11] Saman Shishehchi, Seyed Yashar Banihashem, Nor Azan Mat Zin,"A Proposed Semantic Recommendation System for E-Learning," Information Technology (ITSim),Vol.1, pp. 1-5, 2010.

[12] T. Berners-Lee, J. Hendler and O. Lassila, "The Semantic Web," Scientific American, May 2001.

[13] Xinye Li, Qinhai Yang, LinNa Zeng, "Clustering Web Retrieval Results Accompanied by Removing Duplicate Documents," Web Information Systems and Mining (WISM), Vol.1, pp.259-261, 2010.