Enhancing Reasoning and Retrieval Performance on OWL using SQWRL

[1]A. A. Obiniyi, [2]O. N. Oyelade, [3]S. B. Junaidu

[1,2,3]Department of Mathematics, Ahmadu Bello University
Zaria, Kaduna State, Nigeria.

***Abstract*--The Semantic Web Initiative advanced the web from a syntactically oriented web to semantically oriented web. Different ontology languages are used in modeling information about specific domain in the Semantic Web and the choice of this ontology language will influence the degree of expressivity of concepts of such domain. However, the choice of ontology language does not affect the representation of the model alone, but as well the efficiency of the chosen Semantic Web enabled query language that is to be used on the desired ontology language. Furthermore, the Semantic Web also enables ontology developer to implements rules that will foster ontological reasoning upon the underlying ontology. This research seeks to demonstrate how performance of reasoning and query output can be improved using Web Ontology Language (OWL) as ontology language used to model the concepts of a domain of choice. This was chosen from other ontology languages such as Resource Description Framework/ Resource Description Framework Schema (RDF/RDFS), due to it's in depth expressivity power. Also, Semantic Query-enhanced Web Rule Language (SQWRL) is chosen from among other Semantic Web enabled query languages, such as OWL Query Language (OWL-QL) and SPARQL RDF Query Language (SPARQL), due to its feature of rightly interpreting the semantics of OWL constructs. An implementation of this reasoning and retrieval performance was further illustrated by the application developed in the research work.**

Keywords: Semantic Web, OWL, RDF, RDFS, OWL-QL, SQWRL, SPARQL and Reasoning

I INTRODUCTION

The World Wide Web (www), also known as the web, has been greatly altered by the semantic web. Some of the peculiarities of the initial web before the semantic web are its simplicity in terms of data or information display. Data provided on this web have been in the format of Comma Separated Values (CSV), eXtensible Markup Language (XML), and Hyper Text Markup Language (HTML) tables, sacrificing its semantics for syntax (Bizer*et al*, 2009). Search engines under the initial web were syntactically driven- searches are carried out based on the word match of both the information in the database and that of search words entered by the user- however; the semantic web carries out its search on the web based on the semantic of the search word entered by the user.This is in line with the original vision of the semantic web initiative, which states that machine or computer will be able to analyze or process data on the web (Berners-Lee, 1994).In pursuit of this vision, the Semantic Web and its constituent's technologies have become the *de factor* in realizing it. Some associated technologies of the semantic web can be categorized into the ontology languages, such as OWL and RDF/RDFS, querying languages, such as SPARQL, OWL-QL and SQWRL, and rule languages, such as Semantic Web Rule Language (SWRL). Rule engines like Jess(http://www.jessrules.com/jess/index.shtml) and Reasoning engines like Racer(http://www.sts.tu-harburg.de/~r.f.moeller/racer/) and Pellet (www.mindswap.org/2003/pellet/index.shtml).Ontology modeling is done to explicitly model formal conceptualization about a domain of interest, and the model ontology is supposed to be used and understood by machines for the purpose of knowledge sharing. Modeling such knowledge bases demands that ambiguity of specifications be avoided. Also, it demands that machines should be able to make inferences from such knowledge just as what the semantic web was developed to achieve. Hence, the choice of an ontology language must be carefully weighted so that concepts can be correctly modeled using them. Several ontology languages abound with the semantic web, and some of which are OWL and RDF/RDFS. But to be able to retrieve by inference information that may not be readily modeled in the underlying ontology, OWL more powerful and expressive features or constructs that provides ontology modelers with richness of expressivity that engenders reasoning or inference making. Comparing OWL and RDF, it is noted that RDF/RDFS permits some ontological knowledge representation and its modeling constructs concerns with the organization of vocabularies in typed hierarchies: subclass and sub property relationships, domain and range restrictions, and instances of classes.

However, some limitations known with it are with respect to lack of more constructs that will provide intuition. Some representation that cannot be realized with RDF/RDFS are; *disjointness* of two different concepts, a boolean combination of two or more classes or concepts of adomain,

specifying restrictions on how many distinct values a property may or must take, specially characterizing properties and localizing the scope of properties or relations of classes are not all available in the features of RDF/RDFS (Antoniou*et al*, 2008). On the contrary, OWL provides constructs to combat these deficiencies of RDF/RDFS, making OWL a more expressive and powerful ontology language that helps in building ontology, state facts about a domain, and as well realizing reason among ontologies and facts.OWL use the close world assumption which states that [The closed] world assumption implies that everything that is not known is false, while the open world assumption states that everything that is not known is *undefined*(Mazzocchi, 2005). OWL facilitates greater machine readability of web content than XML and it also extends RDF/RDFS by providing additional vocabulary along with a formal semantics. It has built-in ontology mapping support (equivalent classes, and *sameAs*), some other property types (symmetric, transitive, functional) and it allows for the definition of classes by the attributes of their members (enumeration, restrictions and logical statements) (Knublauch*et al*, 2003). All these features add up to the power of expressivity of OWL.
Writing ontology by hand or manually can be quite demanding and error prone especially when more complex conceptualization is to be engineered. Therefore, ontology development tools such as Protégé were developed for this purpose. Protégé is an extensible and customizable toolset constructing ontologies and for developing applications that use these ontologies. Outstanding features that they provides includes; automatic generation of graphical-user interfaces, based on user-defined models (Knublauch*et al*, 2003), provision for plugins that are used for reasoning and rule implementation mechanism. Rules are implemented or added to ontologies using rule languages. SWRL is a rule language that is used to add rules to OWL ontology. It was developed as a formal description logic-based extension to OWL and provides an expressive language that is strongly suited to types of changes or additions that is required when performing knowledge mapping. OWL and SWRL are core semantic web languages (O'Connor *et al*, 2010). Rules formed using SWRL are added into the underlying ontology using rule engine like Jess.

Reasoning over ontology requires reasoning engines like Pellet and Racer. Pellet provides reasoning that is sound and complete for a variant of OWL called OWL Description Logic (OWL DL) and has extensive support for reasoning with individuals, user-defined datatypes, and debugging support for ontologies (Sirin*et al*, 2007). When rules are added to ontologies, they provides room for inference making on the ontology, also, when a reasoner reasons over an ontology, it derives facts or premises that were inference from the underling ontology. Sometime, rule engine and reasoning engine can be combined and used on a specific ontology. For example, Protégé provides a tab called SWRLJessTab which combines the Protégé OWL Plugging, Racer and Jess, where Racer processes OWL DL and Jess executes production rules (Mei, 2005).When some of these semantic web technologies are combined, a very good application which is semantically oriented can be developed. One of such applications is a work by Shamoug*et al*(2012). The research examined the possibility of embedding ontological reasoning as tool for humanitarian crisis decision making. An ontology whose concepts were derived from the possible actions and responses (such as WHO, WHAT, WHERE, HOW, WHEN) about the occurrence of an event or humanitarian crisis, was developed. Then SWRL was used to formulate rules that were used in the decision making. This application will be able to take relevant decision for decision makers based on the knowledge base model in the ontology.This paper looks at the possibility of exploiting efficient query language in retrieving information from OWL models.

II.SQWRL AND OTHER SEMANTIC WEB QUERY LANGUAGES

There quite some query and rule languages that are used alongside ontologies. In this section, some of these languages are going to be briefly discussed with regards to their relation with SQWRL.

A. SQWRL and other Query Languages

The performance of data retrieval in any data management system or model is of paramount concern. Options like query optimization have been exploited for some of the relational databases query languages like Sequel Query Language (SQL) in other to optimize performance. Semantic query provides users to tune a collection of semantic parameter to formulate an intended request (Bouquet *et al*, 2004). Different query languages are also available for use in ontological query. Some of these are SPARQL, OWL-QL, asking (ASK) and SQWRL. Some of these query languages have their deficiencies which eventually affects either the result of query or their understanding of entailment in the underlying ontology. For instance, OWL-QL is based on OWL and is used to manage request-response dialog thereby creating a communication rule for both procedures. It serves as request-response agents that seekanswers to web service queries(Bitar*et al*,2011).ASK/TELL isa query language that is formulated for ontological knowledge base query, and it is a research work of Description Logic Group (DIG). The query is effectively XML schemas for a description logic concept language, providing the functionality of ASK/TELL and it provides interfacing with some prominent DL reasoners such as FaCT and RACER (Bechhofer, 2003). On the contrary, no practical implementations of OWL-QL were made available and the ASK protocol is too inexpensive to be used as a general OWL query language (O'Connor *et al*, 2009). SPARQL is another ontology querying language that is traditionally used with RDF and this limited its effectiveness (Harbelot*et al*, 2013) and it does not have a native understanding of OWL's semantics, as a result, it cannot directly query entailments made implemented using OWL constructs (O'Connor *et al*, 2009).

B. SQWRL and Rule Language

Knowledge bases implemented with ontology language like OWL are further enriched by adding rules into them in other that entailments or inferences may be derived from such ontology. For instance, it is possible to use OWL to specify that every male is a human being. With the aid of rules, a new specification implying that every human older than eighteen years is an adult could be derived. Semantic Web Rule Language (SWRL) permits OWL users to write rules that can be expressed in terms of OWL classes and that can reason about OWL individuals (O'Connor *et al*, 2010) and one of its powers lies in ability to support built-ins predicates defined by users which are usable in SWRL rules. Hence, enhancing reasoning or inference making over OWL ontology is made possible by SWRL.

C. SQWRL in a Nutshell

SQWRL is a SWRL-based query that provides some form SQL-like operators that can be used to construct query that will retrieve information from OWL. It is built on the SWRL rule language and assumes the standard SWRL rules antecedent making out a query or pattern specification for retrieving information from OWL and standard SWRL serialization mechanisms can be used so queries can be stored in OWL ontologies (O'Connor *et al*, 2009).One other feature provided by SQWRL is its ability to support querying information sourced from different ontological knowledge bases. This gives room for enriching querying out information from a semantic interoperable data sources and SQWRL can be enriched semantically to deal with semantic heterogeneity.SQWRL is composed of two types of elements which are application ontology elements (classes and properties) and values (constants).It also has different operators or constructs that are used in making up a desired pattern specification-query – some of which are the ***select***, ***count***and***orderby***, and aggregation construct such as ***max***, ***min*** and ***avg***. when these features are well used in querying ontology, the result of the query can be greatly enriched.

SQWRL takes on the left hand side or antecedent of a SWRL as the query specification and then used the own construct on the coincident part or the right hand side of the query. The left hand side of the query is composed of atoms which in themselves are a combination of concepts of ontology or properties and variables or constants. For instance, to query ontology about products (for example, mobile phones product)we could write a query such as

MobilePhone(?m) ->sqwrl:select(?m) …(1)

The result of this query is a list of classes or concepts, individuals or instances and other entailed concepts about ***MobilePhone*.**The query (1)has just one atom with one predicate and variable on the antecedent part of the query. Query (1) can further be modified to retrieve more specific result. Query (2) adds this improvement.

MobilePhone(?m) ^ hasPrice(?p) ^ swrl:lessThan(?p,N5000)->sqwrl:select(?m) ..……………………(2)

The query (2) will retrieves all mobile phone products and their prices, but whose prices must be less than N5000. The atom *swrl:lessThan(?p, N5000)* makes use of SWRL built-in. This interoperability of SQWRL and SWRL constructs helps to improve the formulation and performance of the query and as well create space for accessing entailments added through into the ontology through SWRL. This interoperability of SWRL and SQWRL is not obtainable with other query semantic languages like SPARQL and OWL-QL. One relevant difference between SPARQL and SQWRL is that the count operator in SPARQL find out the number of rows in the underlying ontology that have to do with the concept or class mentioned in the query while that of SQWRL counts the number of rows that are found in the result of query. For instance, (3) will return the number of mobile phones whose prices are less than N5000 whereas SPARQL would have counted the number of mobile phones in the ontology.

MobilePhone(?m) ^ hasPrice(?p) ^ swrl:lessThan(?p,N5000)->sqwrl:count(?m)…………………………(3)

All these features in SQWRL empowers it's to improve both the performance of reasoning and querying of ontology. When executing SQWRL, the SWRL inference rules is first executed and then the SQWRL part of the query is then executed. Implementing or running SQWRL queries requires a rule engine and currently Jess is used (Necula, 2012).

III. MATERIALS AND METHODS

In this section, first the architecture of the system implemented for thispaper is explained. Figure 3.1 shows the system architecture. The architecture reveals that there is the client side and the server side. The client uses a semantic standalone application to make connection to the server side. The server side authenticates the user before processing the search inputs against two categories on ontology. The two category of ontology that the server side deals with is the Core ontology and the Retailer ontology. The Core ontology models information about online product retailers/sellers who deals with a particular product. This ontology captures information such as their ontology link address and other contact information about them. From this Core ontology, the search for the desired product of the user can be further redirected to the personalized ontology of the retailer. The result of the search is formatted for output for display on the user's application.

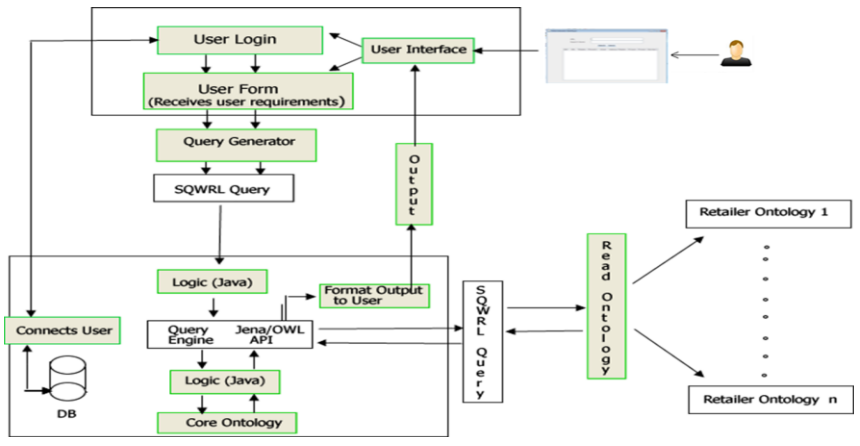


Figure 3.1 System Architecture

The two ontologies files designed for this paper is as shown in Figure 3.2and 3.3. OWL was the choice ontology language used to model the concepts in these two domains. Generally in OWL, information or a fact can mean and be understood differently by query languages based on the level of expression and the type of OWL construct used to model the ontology. For instance, looking at Figure 3.2, the part of the ontology captured was the definition of a concept or class called *Retailer*. It was specified that this class contains some other facts thatcould aid inference making or entailments. The OWL *Restriction*construct is used to achieve this, and it implies that *a Retailer can be drawn from the available categories of Retailers*, that is, one could have retailers that deal with books, while others deals with mobile phone devices- all these are different categories of retailer that a certain retailer can be drawn from.

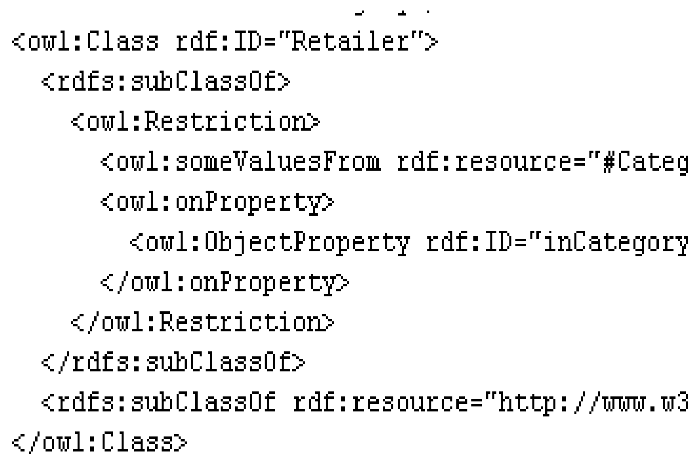

```
<owl:Class rdf:ID="Retailer">
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:someValuesFrom rdf:resource="#Categ
      <owl:onProperty>
        <owl:ObjectProperty rdf:ID="inCategory
      </owl:onProperty>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf rdf:resource="http://www.w3
</owl:Class>
```

Figure 3.2 Core ontology model

Also, the OWL constructs *ObjectProperty*, is used to establish a relation between the two concepts – *Retailer* and *Category*- which is read as, *Retailer inCategory Category*. This implies that the instances or individuals that of both classes *Retailer* and *Category* are going to be retrieved once they have the relations or predicate *inCategory* on them. Now to formulate SQWRL query that will do these, this could be written like;

Retailer(?r) ^ inCategory(?r, MobilePhones) ->sqwrl:select(?r)
…………………………………………….(4)

Query (4) will select all retailers that deals with mobile phones. However, since the user will want to visit the online store of such retailers, it will be important to require that their website address be retrieved also. Then query (4) can then be modified to include this addition

Retailer(?r) ^ inCategory(?r, MobilePhones) ^ hasURL(?r, ?url)
->sqwrl:select(?r, ?url)……………………………(5)

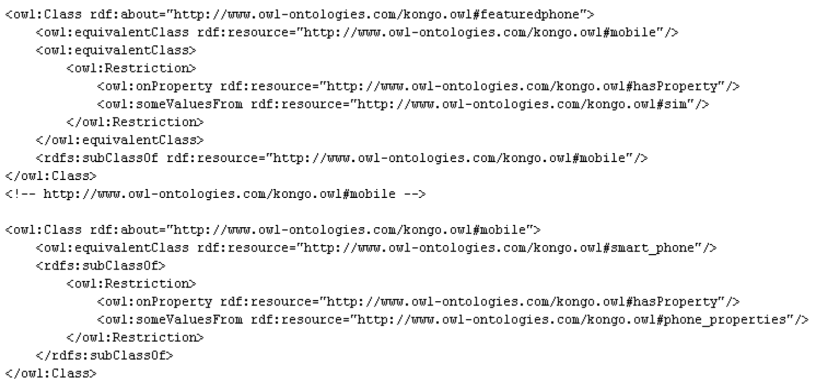
SWRL can be used to formulate a rule that will be added to the ontology so that subsequent query for the same result of (5) will be an entailment that can be easily retrieved from the ontology. This can be further changed to;

Retailer(?r) ^ inCategory(?r, MobilePhones) ^ hasURL(?r, ?url)
->allMobilePhonesRetailer(?r,?url)....r1

Rule r1 is SWRL rule which when implemented using a rule engine – Jess – it will add that to the ontology and SQWRL can then be used to query this inference or entailments by taking the right hand side of the rule as its left hand side and then the select construct of SQWRL will appear on the left hand side. This is shown by query (6).

allMobilePhonesRetailer(?r, ?url)->sqwrl:select(?r,?url)
……………….……(6)

Figure 3.3 captures the modeling of the retailer domain.



```
<owl:Class rdf:about="http://www.owl-ontologies.com/kongo.owl#featuredphone">
    <owl:equivalentClass rdf:resource="http://www.owl-ontologies.com/kongo.owl#mobile"/>
    <owl:equivalentClass>
        <owl:Restriction>
            <owl:onProperty rdf:resource="http://www.owl-ontologies.com/kongo.owl#hasProperty"/>
            <owl:someValuesFrom rdf:resource="http://www.owl-ontologies.com/kongo.owl#sim"/>
        </owl:Restriction>
    </owl:equivalentClass>
    <rdfs:subClassOf rdf:resource="http://www.owl-ontologies.com/kongo.owl#mobile"/>
</owl:Class>
<!-- http://www.owl-ontologies.com/kongo.owl#mobile -->

<owl:Class rdf:about="http://www.owl-ontologies.com/kongo.owl#mobile">
    <owl:equivalentClass rdf:resource="http://www.owl-ontologies.com/kongo.owl#smart_phone"/>
    <rdfs:subClassOf>
        <owl:Restriction>
            <owl:onProperty rdf:resource="http://www.owl-ontologies.com/kongo.owl#hasProperty"/>
            <owl:someValuesFrom rdf:resource="http://www.owl-ontologies.com/kongo.owl#phone_properties"/>
        </owl:Restriction>
    </rdfs:subClassOf>
</owl:Class>
```

Figure 3.3 Retailer ontology model

Concepts such as featured phone, mobile phone, smartphone, and SIM were used to model the domain and predicate such as *hasProperty*is used to relate some of the concepts together. Here, after the desired retailers have been retrieved by the query in (5) or (6), then the semantic searcher can further query for the request of the user. For instance, if the user enters a query like *I want a Nokia dual sim phone*, the semantic application will carry out the tokenization process to retrieve appropriate words for formulating a query after which a as shown in (7) will be executed.

Product(?product) ^ hasProperty(?product, dual) ^ hasName(?product,Nokia)->sqwrl:select(?product, ?amount) ……………………………………………...(7)

However, getting information about a product is not sufficient if the price of the product is not known. Moreso, the user may choose to add some constraint by saying that only such products gotten by query (7) and whose price is less than or equal to N5000 should be queried. Then query (8) can formulate to do this.

Product(?product) ^ hasName(?product, Nokia) ^ hasProperty(?product, dual) ^ hasPrice (?product, ?amount) ^ swrlb:lessThanOrEqual(?amount, N5000)

*->sqwrl:select(?product,?amount)…….(*8)

Query (8) shows how SWRL built-in *swrlb:lessThanOrEqual*was been employed to evaluates the constrain user placed on the price. Finally, if consideration is given to a situation where the user demands for the number of such products retrieved by query q8, then the SQWRL feature or construct *sqwrl:count*can be used to archive this as shown in query (9).

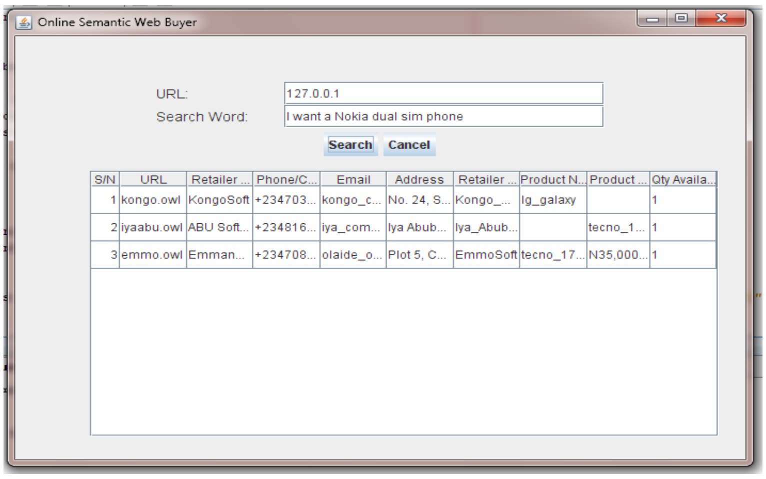
Product(?product) ^ hasName(?product, Nokia) ^ hasProperty(?product, dual) ^ hasPrice (?product, ?amount) ^ swrlb:lessThanOrEqual(?amount, N5000)

*->sqwrl:select(?product,?amount) ^ sqwrl:count(?product) ……………………(*9)

The count operator is a deficiency on the side of SPARQL in terms of counting the result of a particular variable in the query result, thereby putting SQWRL ahead of it. Some more expressive SQWRL constructs captured in Figures 3.2 and 3.3, together with its capability to work seamlessly with a SWRL rule language makes it more proficient in executing efficient queries as well to enhances reasoning or inference making.

IV. RESULTS

To demonstrate the illustrations carried out in section 3.1, an implementation for the semantic web application was done alongside this research and this application was further tested against the ontologies mentioned in section 3.1. The implementation was developedas standalone application that provides users with connectivity with the server where the user's query request is first used to locate the relevant retailer ontology to navigate the request to. The output of the query is displayed in the application as shown in Figure 4.1. The user can then scroll through the result of the query and then choose to visit the website of a desired retailer. Again here it is assumed the user entered the query: *I want a Nokia dual sim phone*. Based on these search request, three rows of information where retrieved.

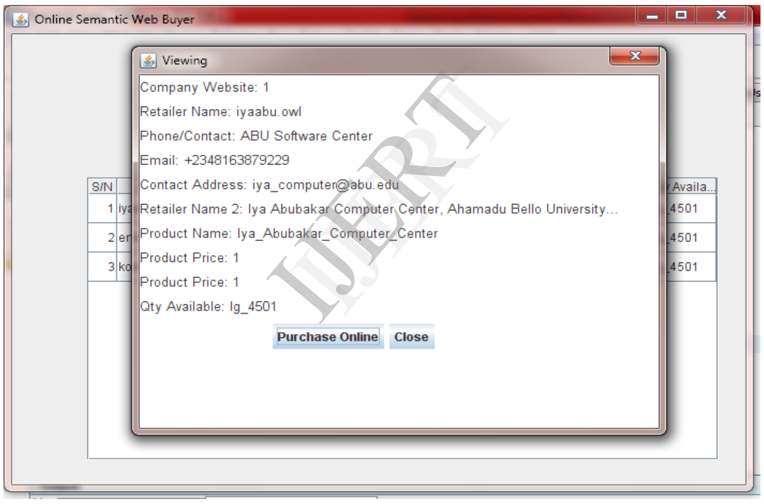


URL: 127.0.0.1
Search Word: I want a Nokia dual sim phone

S/N	URL	Retailer ...	Phone/C...	Email	Address	Retailer ...	Product N...	Product ...	Qty Availa...
1	kongo.owl	KongoSoft	+234703...	kongo_c...	No. 24, S...	Kongo_...	lg_galaxy		1
2	iyaabu.owl	ABU Soft...	+234816...	iya_com...	Iya Abub...	Iya_Abub...		tecno_1...	1
3	emmo.owl	Emman...	+234708...	olaide_o...	Plot 5, C...	EmmoSoft	tecno_17...	N35,000...	1

Figure 4.1 User request entered in application

When a user's request is processed and the eventual result displayed, based on the retailer information the user chooses, a dialog box will popped up to give the user a breakdown of the chosen retailers information which will be shown in the dialog box and the user will be requested to either visit the online shop of the retailer or close the dialog box to choose another retailer. Figure 4.2 captures this illustration.



Company Website: 1
Retailer Name: iyaabu.owl
Phone/Contact: ABU Software Center
Email: +2348163879229
Contact Address: iya_computer@abu.edu
Retailer Name 2: Iya Abubakar Computer Center, Ahamadu Bello University...
Product Name: Iya_Abubakar_Computer_Center
Product Price: 1
Product Price: 1
Qty Available: lg_4501

Figure 4.2 User viewing a retailer's product information

This implementation clearly sidelines the synthetic search engines that have become *de factor* over time in searching out specific information from an innumerable number of information linked together on the Web. This application is not synthetic based but rather a semantic web based application that can understand or read meaning into the search request of the user and going directly into relevant online data in other to retrieve query result for the user.

V.CONTRIBUTION AND RECOMMENDATION

Data retrieval is a major operation been carried out on data models. However, in Semantic web data modeling; provision for reasoning is also made available through the combination of both the rule and query languages. Hence, this paper have demonstrated the use of SQWRL as a query language in retrieving information from OWL model, in addition, SWRL , a rule language was further used alongside SQWRL in a bid to enhance intelligent retrieval operation. However, dynamic way of using SWRL to populate or increase the information in the underlying ontology can be added to this work. This will enable the ontology developer to do less in improving the information in the ontology.

VI. CONCLUSION

In this paper, the rich features of SQWRL have been explored. The query itself has been compared with other query languages in other to show that it understands the semantics of OWL more than them. Furthermore, different ontology languages such as RDF and OWL were discussed, but OWL was pointed out as ontology language with rich expressivity compared to others. SWRL can be used to enrich OWL ontology. The rule resulting from this SWRL will form individuals or instances that are derivable from the ones in the underlying ontology. Also, it was noted that SQWRL understands rules written in rule language SWRL and this compatibility between it and SWRL makes it feasible to effectively reason over some entailments that ordinarily may not have beenretrievable by other semantic query languages. An implementation of these arguments was also demonstrated and the result was shown in Figures 4.1 and 4.2

REFERENCES

1. Antoniou, G., and Frank, V. H., (2004). *Semantic Web Primer* MIT Press, Cambridge, pp. 115-116.
2. Bechhofer, S., (2003). The *DIG Description Logic Interface: DIG/1.1*, University of Manchester, Manchester, pp. 2.
3. Berners-Lee, T., (1994). *Weaving the Web*, The MIT Press Bookstore, Cambridge, pp. 9.
4. Bitar, I. E., Belouadha, F., Roudiès, O., (2011). *Taxonomy and Synthesis of web Services Querying Languages*, International Journal of Science and Advanced Technology (ISSN 2221-8386) Volume 1 No. 4, pp. 43
5. Bizer, C., F., Heath, T., T., and Berners-Lee, T., (2009). *Linked Data-the Story so Far*,Universität Berlin, Germany, pp. 1.
6. Bouquet, P., Kuper, G., Scoz, M., and Zanobini, S., (2004). *Asking and Answering Semantic Queries*, University of Trento, Via, Sommarive, Italy, pp. 1.
7. Harbelot, B., Arenas, H., and Cruz, C., (2013). *A Semantic Model to Query Spatial-Temporal Data*, The 6th International Workshop on Information Fusion and Geographic Information Systems: Environmental and Urban Challenges, St. Petersburg: Russian Federation, pp. 3
8. Jess. Retrieved October *22, 2013 from*http://www.jessrules.com/jess/index.shtml
9. Knublauch, H., Musen, A. M., and Noy, F. N., (2003). *Creating Semantic Web (OWL) Ontologies with Protégé*, 2nd international Semantic Web Conference (IJWC2003), Florida, USA, pp. 8-9.
10. Mazzocchi, S., (2005). *Closed World vs. Open World: the First Semantic Web Battle*. Retrieved November11*, 2013 from*http://www.betaversion.org/~stefano/linotype/news/91/
11. Mei, J., (2005). *Reasoning Paradigms for SWRL-enabled Ontologies*, Department of Information Science, Peking University, pp. 3.
12. Necula, S.B. (2012). *Implementing the Main Functionalites Required by Semantic Search in Decision-Support Systems,* International Journal of Communication, ISSN 1841-98367(5):907-915, pp. 908.
13. O'Connor, M. J., Shankar, R., Nyulas, C., Tu, S., and Das, A., (2010). *Developing a Web-Based Application usinf OWL and SWRL*, Stanford Medical Informatics, Stanford University, Stanford, pp. 1-2.
14. O'Connor, M., and Amar, D. (2009). *SQWRL: a Query Language for OWL*, Stanford Center for Biomedical Informatics Research, Stanford, pp. 1.
15. Pellet. Retrieved November 5 2013 from www.mindswap.org/2003/pellet/index.shtml
16. Racer. Retrieved November 5 2013 from http://www.sts.tu-harburg.de/~r.f.moeller/racer/
17. Shamoug, A., and Juric, R., and Paurobally, S., (2012). *Ontological Reasoning as a Tool for Humanitarian Decision Making*, Proceedings of the 9th International ISCRAM Conference, Vancouver, Canada, pp. 1.
18. Sirin, E., Parsia, B., Grau, B. C., Kalyanpur, A., and Katz, Y., (2007). *Pellet: A Practical OWL-DL Reasoner*, University of Maryland, pp. 1