

## RDF and OWL Ontology Building of Web Applications

Shabana Samreen, J.S. Mirza and Anila Rasheed

Department of Computer Science, COMSATS Institute of Information Technology Lahore, Pakistan

**Abstract:** The goal of semantic web is to understand human language with meaning. Semantic web include metadata that is called annotation. RDF and OWL both languages have standard formats for the sharing and understanding of knowledge on the web. Ontologies also originate from Semantic Web. These RDF and OWL Ontology for web application like Library System of CIIT (Comsats Institute of Information Technology) of Lahore campus can be built. With the help of this ontology we can search the data, understand and customize the data, efficient reasoning support. These OWL ontologies include classes, properties, individuals and their instances. How these classes relate with each other. All the properties related to the Book Information System have set into the data type properties, object properties and annotations. In this research study we proposed the work related to RDF and OWL Ontologies comparison between RDF and OWL ontology and XML schema relationship between the cases of this ontologies, how this ontology was helpful in our web applications. We provide the best way of building of ontology for any web application and any other organization. When the ontology is build, result is obtained by the SPARQL queries with efficient reasoning support. This ontology is developed for CLS (Comsats Library System) in Lahore campus. Manipulation of different properties and then binding of protégé with eclipse to give support for user interface is important task. The results are best correctly gained through SPARQL query.

**Keywords:** Jena API, ontology building, OWL, protégé, RDF, semantic modelling, SPARQL query, web application, XML

## INTRODUCTION

Comsat has a number of campuses that are all over spread in Pakistan. It has seven campuses in Pakistan. We have taken just case of Lahore campus. In the Lahore campus, there are many departments and programs. Library system has very important factor in any education system. This is difficult for students to search library system. In the semantic web, the search engine does not give accurate result it provides many results that are not much more accurate as like the user wants. Liyang (2011), John *et al.* (2009) and Toby *et al.* (2009). But in the semantic web it gives us the accurate and synonymous result. The user may not have enough time to search separate hyperlinks that are generation in the traditional web. This difficulty can be solved by building ontology with the help of protégé tool that is open editor tool (Natalya and McGuinness, 2009; Jonathan, 2004; Michael Grobe, SIGUCCS, 2009).

The traditional web search engine is designed for human not for machine. For example, when we enter some key words in the traditional web search engine like Google, just like "Give name of Web Semantic Books by author BobDucharme". In the Google, It give us the many results related to book but not more accurate result that contained many web site. We need

[Learning SPARQL: Bob DuCharme: 9781449306595: Amazon.com ...](#)  
[www.amazon.com > ... > Computer Science > Information Theory](#) ▼  
Learning SPARQL [Bob DuCharme] on Amazon.com ... List Price: \$29.99. Price ...  
With this concise book, you will learn how to use the latest version of this W3C ...

[XSLT Quickly: Bob DuCharme: 9781930110113: Amazon.com: Books](#)  
[www.amazon.com > Books > Computers & Technology > Networking](#) ▼  
XSLT Quickly [Bob DuCharme] on Amazon.com ... Bert Blevins Web Design and  
Interface Development Books: A list by Bert Blevins "Bert Blevins SharePoint" ...

[Bob DuCharme Books List, Complete Bob DuCharme Bibliography ...](#)  
[www.ranker.com > Books > Author Bibliographies](#)  
Bob DuCharme books list. Bob DuCharme bibliography includes all books by Bob  
DuCharme. Book list may include collections, editorial contributions, etc - any ...


[Bob DuCharme: home \(page\) - Snee](#)  
[www.snee.com/bob/](#) ▼  
Free epub children's picture books These were developed with the OLPC XO laptop ...  
XSLT 2.0 articles: XML.com has a complete list of my Transforming XML ...

[Amazon.co.uk: Bob DuCharme: Books](#)  
[www.amazon.co.uk/Books/s?...Bob%20DuCharme...27%3ABob%20Du...](#) ▼


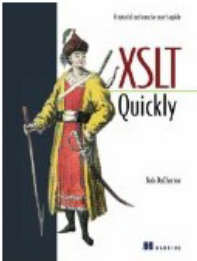
Fig. 1: Searching a book by Google

to go the linked websites and search the related book of Semantic Web that has the author name Bob Ducharme. This is shown in the Fig. 1.

When we want to search same thing in the Semantic Web. This gives us the accurate result and gives all the Semantic Web Books by Bob Ducharme. The Semantic Web also gives the related

1.  **Learning SPARQL** by Bob DuCharme (Jul 23, 2011)  
★★★★★ (9)  
**Formats**

	Price	New	Used
<b>Paperback</b>	Order in the next <b>34 hours</b> to get it by Tuesday, May 21. Eligible for FREE Super Saver Shipping.	<del>\$29.99</del> <b>\$24.01</b>	\$20.11 \$20.00

[Sell this back for an Amazon.com Gift Card](#)
2.  **Amazon's Bob DuCharme Page**  
Discover books, read about the author, find related products, and more. [More about Bob DuCharme](#)  
**Bestselling Books:** Learning SPARQL, XSLT Quickly, The Operating Systems Handbook.
3.  **XSLT Quickly** by Bob DuCharme (Jan 1, 2001)  
★★★★★ (13)  
**Formats**

	Price	New	Used
<b>Paperback</b>		\$100.00	\$5.33

Fig. 2: Searching a book by web semantic search engine

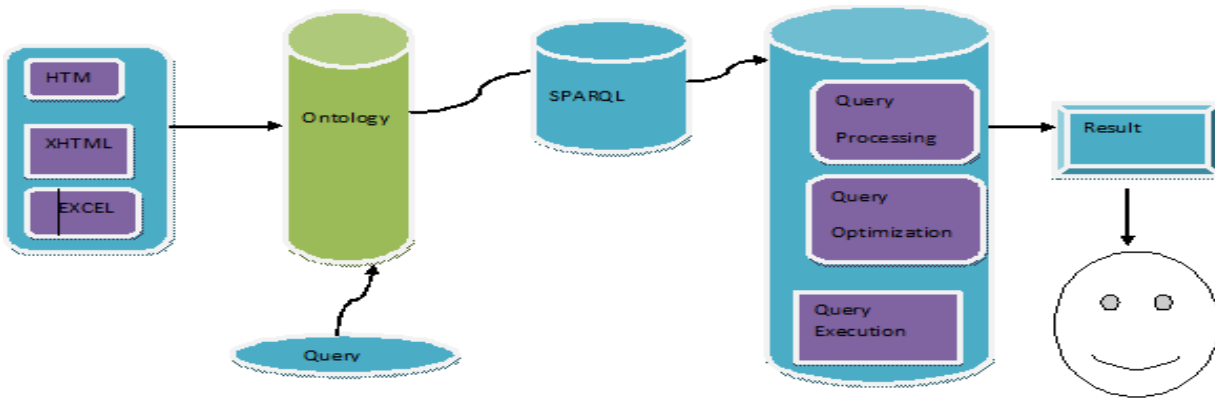


Fig. 3: Architecture

information to the Semantic Web Books of Bob Ducharme like price, title and date of publication (Fig. 2).

Semantic web provide proof and trust to the user because it give us the accurate and proof result to user. SPARQL query is also used to retrieve the data from the web semantic languages like RDFs, RDF and Owl. (Liyang, 2011; John *et al.*, 2009; Toby *et al.*, 2009) This SPARQL query can be generated and executed on different tools like Protégé. The SPARQL query when generated and after execution it give result, it involve three steps like Query Processing, Query optimization

and Query execution (Sven, 2011; Aditya *et al.*, 2006) (Fig. 3).

We build the ontology of web application like Comsats Institute of Information Technology Lahore. We build the ontology of both programs like SDP (Single Degree Program) and DDP (Dual Degree Program).

## MATERIALS AND METHODS

Semantic is just like as WorldWide Web. It helps the people to gain knowledge beyond many web

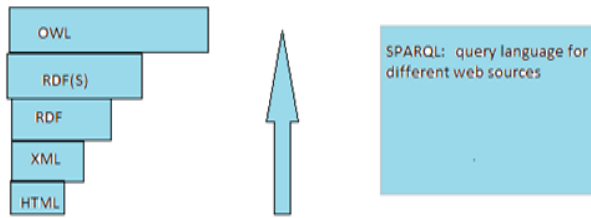


Fig. 4: Web semantic languages

applications. In the World Wide Web when we want to search something, this is not necessary to understand the meaning of world that we want to search. But in the Semantic Web this is not possible to understand. Semantic Web is just study of meaning. When we want to search something in the Semantic Web, it understands the meaning of the word. In the domain of computer Science, semantic builds relationship between syntax and its interpretation. OWL documents that we build in the language of OWL are also called Ontology. OWL is enhancement of RDF and RDFs. RDF and OWL is much better than other web semantics languages like XML, UML etc. The RDF and OWL ontology is build with the help of Protégé. McBride *et al.* (2010), Aditya *et al.* (2006) and Malik *et al.* (2010) There is strong relationship between the RDF and OWL. Without RDF, the classes of OWL cannot declare. This is impossible to make OWL ontology without using the RDF. Protégé is an open source tool to model a schema with their semantics. Protégé can be used to load, edit and save ontologies in the bundle of formats like RDF, XML, UML, RDF PLUS and OWL etc. (Sven, 2011; Toby *et al.*, 2009; Grigoris and van Harmelen, 2004). There is hierarchy build in the Protégé and classes and subclasses are made in this tool. Protégé has the capability of build the ontology with its own user interface. With the help of ontology this is easy to understand the relationship between the classes, relationship between classes and subclasses. Berners-Lee *et al.* (2001) this is easy to build ontology in both logical and properties view and applying the SPARQL query, (Dean and Jim, 2011) we can extract the result from this ontology for our web application.

Web Semantics come in the form of layers. Web semantics has many languages like XML, UML, RDF, RDFs and OWL etc. Semantic Web comes in the form of layers. These layers are given in the Fig. 4. XML is extensible markup language this is easy to understand but this is not much better than RDF and OWL. RDF is Resource description Framework (McBride *et al.* (2010), Aditya *et al.*, 2006; Malik *et al.*, 2010). This language has more capabilities of understanding as compared to XM. It describes and understands the things, places, meaning and relationship between the things. SPARQL is query language. SPARQL stands for SPARQL Protocol and RDF Query Language. The SPARQL language specifies different query variations for different purposes (Dean and Jim, 2011).

OWL is web ontology language. It has more capability of vocabulary understanding. It better understand the properties and relationship between the classes. Grigoris and van Harmelen (2004) our research proposal includes the RDF and OWL languages. We prefer these languages due to the reason that it has the better understanding of vocabulary, cardinality and other many types of properties. OWL also includes three sublanguages. These languages are categorized due to the capability of their understanding, vocabulary and many other prominent features (Berners-Lee *et al.* 2011; McBride *et al.*, 2010; Sana, 2012).

There are many limitations that are present in the RDF and RDFs. But these limitations are not present in the OWL language. Due to that reason we use both RDF and OWL. First reason is that, in the RDF and RDFs we cannot build the range restriction in the classes. Smith *et al.* (2004), Kendall *et al.* (2008) we cannot say in the RDF that the edition of book Web semantic is 1<sup>st</sup> and other books related to computer science may have edition first. But in the OWL language we can say this statement and OWL language understand the ranges and domain of the things. Second reason is that if we want to take the union, intersection and other Boolean function in the classes. John *et al.* (2009), Daconta *et al.* (2009) and James *et al.* (2008), there is limitation that we cannot use union intersection and Boolean function in the classes. But in the OWL we can use the union intersection and Boolean function in the classes. Gómez-Pérez *et al.* (2004) and Dean and Jim, (2011) If we want to get the edition of the two books both books have the edition 1<sup>st</sup>, than in the class by using the union we can take the union of both classes of books. Jonathan (2004), Berners-Lee *et al.* (2001) and Li *et al.* (2007), third reason is that there is missing of property of disjoint in the RDF and RDFs. While in the OWL we can disjoint classes. We can say that Title and author subclass of BOOK are disjoint with each other. Four reasons is that in the RDF and RDFs, special properties cannot declare in the model, while in the OWL, the special properties related to the classes and subclasses can set like function transitive etc (The Future of the Web is Semantic, 2005; Semantic Web Architecture, Year; Paul *et al.*, 2007).

OWL is the extension of RDF. OWL use the properties and classes of RDF and RDFs like (rdfs: Class) etc. OWL language has three subclasses. These subclasses are categorized according to their specific properties (The Future of the Web is Semantic, 2005; Semantic Web Architecture).

These are web applications there is need to define the special properties, disjoint cardinality and use of Boolean function. Due to that reason, for web applications, this is better to use the OWL and RDF both languages (Paul *et al.*, 2007; Michael Grobe, SIGUCCS, 2009; Rubbani, 2007) (Table 1).

Table 1: Special properties of web applications

Properties	RDF	OWL
Imposes rigid structure and contain complex structure	No	Yes
Lager vocabulary understanding	YES but less	Yes
Satisfy meta-data modeling needs	No	Yes
Involved variety of annotation	No	Yes
Easily express relationship between things	No	Yes
Involved annotation framework	No	Yes
Computational realities	No	Yes
Constraint	No	Yes

- First of all we build the ontology by the open source editor tool like Protégé. This tool provides support for building and saving the ontologies in different formats like XML, RDF/RDFS, OWL and RDF Schema etc. Some authors have already been investigated for querying xml data with SPARQL but this query is done on the XML databases. They gain the accurate result by using the SPARQL query but they extract the data from different XML databases not from the XML schema and developed ontology. Various attempts have been made in the literature to address the issue of accessing XML data from within environments. Liyang (2011), Sven (2011) and DuChame (2011), these authors also gain the better results in different semantic environment but not from the XML/RDF schema. Liyang (2011), John *et al.* (2009) and Toby *et al.* (2009), some authors (Toby *et al.*, 2009) produce an XML Schema for HTML search interfaces has been developed. The author provides the XML schema for the HTML search interface but not on the semantic web environment. Some author Herbert (2002) presents a method of generating a representation in form of XML-Schema based on the ontology built with the use of Protégé program but not involved the SPARQL query. Herbert (2002), the question what will be data sources used when we are going to build ontology it may b HTML, XML, RDF/RDFS and involved many other web semantic languages. Than how to translate these languages to other languages (Michael Grobe, SIGUCCS, 2009). We worked and applied the techniques to develop the library ontology of CIIT (Comsats Institute of Information Technology) of Lahore campus. After that we validate our result by using SPARQL and also validate the schema. The scientific contribution to our results are given below in points
- Access the data from the XML schema not from the XML database
- XML schema is produced for the comsats Library
- Use java with help of Jena APIs package to make the user interface not using the HTML interface.
- Goal of our research work is to build interactive and dynamic search engine that is easy to understand by people and give accurate result. This search engine provide accurate result I one line and in less time
- Ontology is build from different data source like HTML, XML and SGML etc. This data is first

purified after normalization. Ontology is build with purified data not from the raw data

- Hierarchical view of library system is build both by manual and by the use of protégé and also their comparison
- First of all we made the RDF/OWL ontology and then may convert this into other languages and other formats like OWL,HTML,CLIPS and N3 etc
- After building of ontology, we describe all related properties that can be used when we build ontology
- At the end we validate and poof our research work by using SPARQL. Query answering is supported by SPARQL (SPARQL Protocol and Query Language) using for OWL embed files. In addition we used Jena APIs package (Li *et al.*, 2007) is used to give support to SPARQL. User interface will transfer these queries by using Jena APIs to the XML document. Because this is dynamic and interactive search engine, this search engine has the advantage that library information from various campuses in the university can be merged. We made the xml document of the web application i.e., Comsat library. After making the XML document, we can make the hierarchy classes of the protégé. This is second way of making formalized and purified data from which we can build ontology.

## RESULTS

Web Semantics involved two main basic languages like XML (Extensible Markup Language) and RDF (Resource Description Framework). The XML document is first created from the different data sources like HTML, XHTML and data from Eexcel. This data is not may in purified form and may contain anomalies and repetition of data. We do first normalize this data. Due to lesser space we do not specify the whole procedure of normalization. After that we prepare the XML document of CLS (Comsat Library System).The XML document is given below:

```
<?xml version="1.0" encoding="UTF-16"?>
<!DOCTYPECiitLhrLibrary SYSTEM
"CiitLhrLibrary.dtd">
<?xml-stylesheettype="text/xsl" href="stylesheet.
xsl"?>
<CiitLahoreLibrary>
  <ComputerEngineering></ComputerEngineering>
  <ManagementScience></ManagementScience>
  <IRCBM></IRCBM>
</CS>
  <ReverseEngineering></ReverseEngineering>
  <ComputerGraphics></ComputerGraphics>
  <ComputerArchitecture></ComputerArchitectur
e>
  <WebSemantics>
    <Book1 title="A Semantic Web Primer" author="
Grigoris Antonious Frank Van Harmeln"
Edition="1st" ISBN="978-0-262-01242-3" Year of
Publication="2008"
```

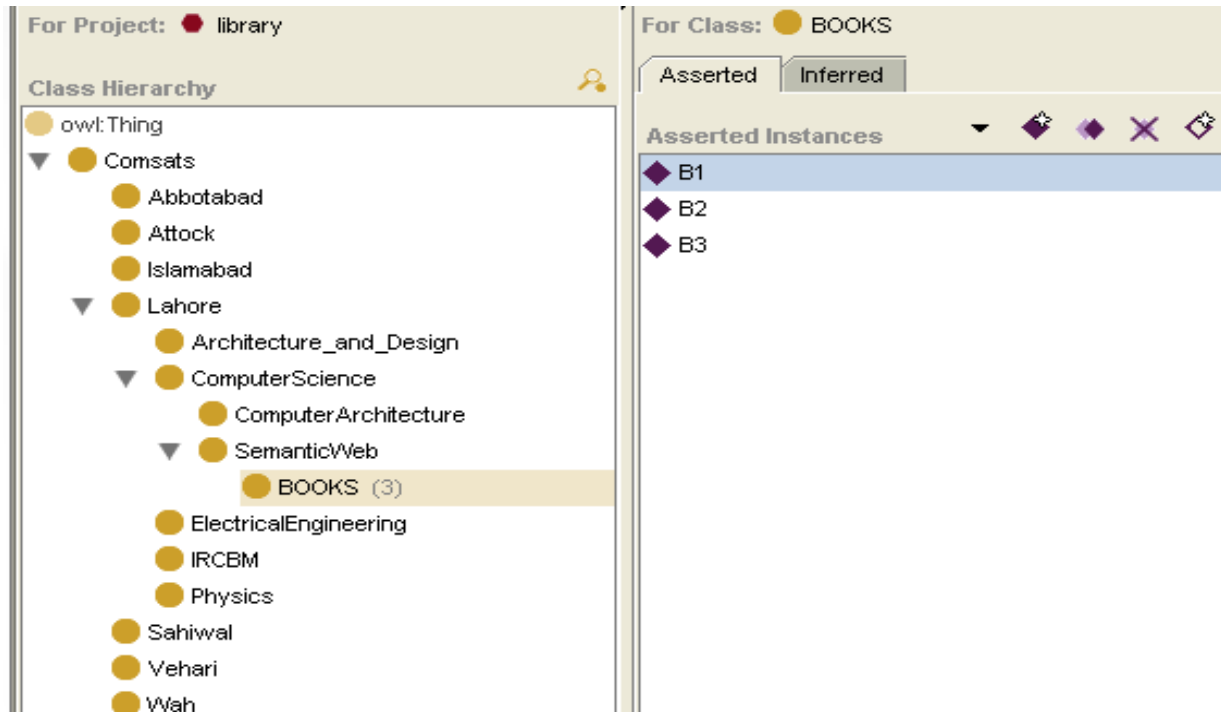


Fig. 5: Individuals of books

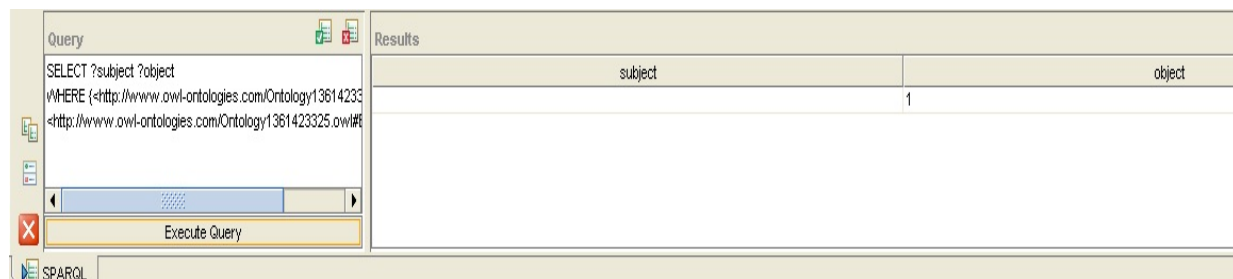


Fig. 6: SPARQL query panel result

```

Published By = "The MIT Press">
</Book1>
<Book2 title = "Learning SPARQL and Updating
with SPARQL 1.7" author = "Bob Ducharme" Edition
= "1st" ISBN = ":978-1-449-30659-5" Year of
Publication = "2011" published By = "O'REILLY">
</Book2>
<Book3 title="Semantic Web for the Word
Ontology Effective Modeling in RDFS and OWL"
author = " DEAN ALLEMANG JAMES HENDLER"
Edition = "1st" ISBN = "13:978-0-12-373556-0"
Year of Publication="2008" Published By =
"MORGAN KAUFMANN" >
</Book3>
</Web Semantics>
</CS>
<Physics></physics>
<Electrical Engineering></Electrical Engineering>
</Ciit Lahore Library>
    
```

When we build on ontology, we must be careful about the classes, subclasses, individuals and property setting. The asserted individual and hierarchical classes of the ontology (Fig. 5).

All the attributes related to book B1 involved the title, edition, ISBN number, Year of publication and published by. These all properties related to the Book are set in the object and data type properties. All properties related to book B1 that are individuals of BOOK are shown as follows:

```

<BOOKS rdf:ID = "B1">
  <AUTHOR rdf:datatype = "&xsd:string" >Frank
  Van Harmeln</AUTHOR>
  <AUTHOR rdf:datatype = "&xsd; string"
  >Grigoris Antonious</AUTHOR>
  <Edition rdf:datatype = "&xsd:int">1</Edition>
  <ISBN rdf:datatype = "&xsd:string" >ISBN978-0-262-
  01242-3</ISBN>
    
```



```
<Publishd BY rdf: data type = "&xsd; string">Mit
Press</Publishd BY>
<rdfs: comment rdf: datatype="&xsd; string"
>All information about B1</rdfs: comment>
<Title rdf: data type = "&xsd; string"> Web
Primer</Title>
<Year of Publication
rdf:datatype = "&xsd;int">2008</Year of Publication>
</BOOKS>
```

If I want to get the edition of Book B1. Than open the Protégé, open the SPARQL query panel when I enter the SPARQL query in this panel, I get the result (Fig. 6).

```
SELECT? subject? Object WHERE {<http://
www.owl-
ontologies.com/Ontology1361423325.owl#B1>
<http://www.owl-
ontologies.com/Ontology1361423325.owl#Edition>?
object}
```

If want to search the author of book B1.The SPARQL query and their result is given below (Fig. 7)

```
SELECT? object WHERE {<http://www.owl-
ontologies.com/Ontology1361423325.owl#B1>
<http://www.owl-
ontologies.com/Ontology1361423325.owl#AUTHOR>?
object}
```

There are many SPARQL queries that are given below to get all related information about the book B1.

- SELECT? object

```
WHERE {<http://www.owl-ontologies.com/ Ontology
1361423325.owl#B1>
<http://www.owl-
ontologies.com/Ontology1361423325.owl#Title>?
object}
```

- SELECT? object

```
WHERE {<http://www.owl-ontologies.com/ Ontology
1361423325.owl#B1><http://www.owl-
ontologies.com/Ontology1361423325.owl # ISBN>?
object}
```

- SELECT? object

```
WHERE{<http://www.owl-
ontologies.com/Ontology1361423325.owl#B1><http://
www.owl-ontologies.com/Ontology1361423325.owl#
Publishd BY>? object}
```

- SELECT? object

```
WHERE {<http://www.owl-ontologies.com/Ontology
1361423325.owl#B1><http://www.owl-
ontologies.com/Ontology1361423325.owl#Year of
Publication>? object}
```

The classes in the ontology show the represented individuals. The classes in the OWL ontology show the root of the element. The classes in the ontology represent the individuals that have common things. This shown the similarity between the groups of individuals:

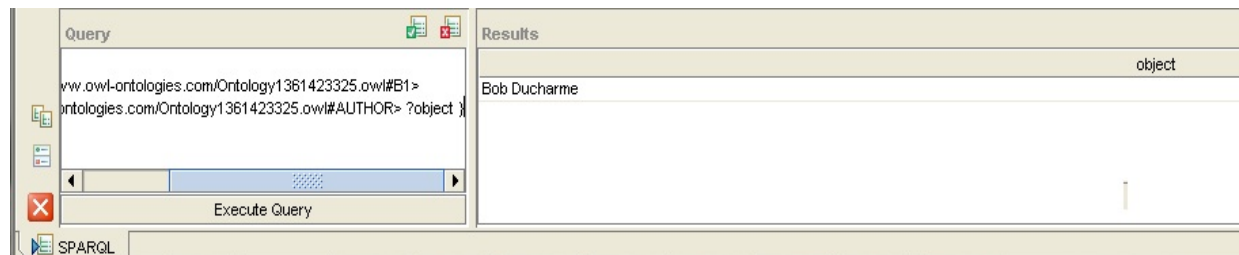


Fig. 7: SPARQL query panel result of author of B1

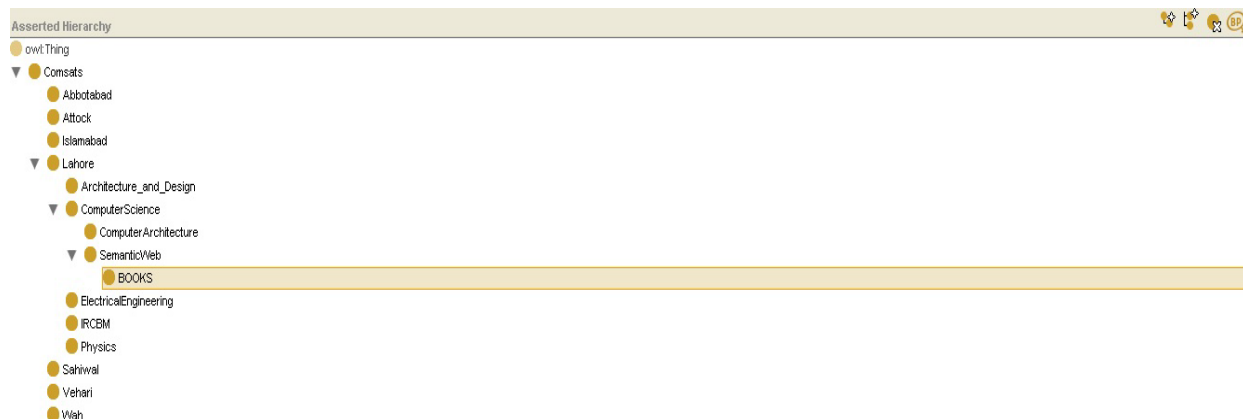


Fig. 8: Hierarchy of classes and subclasses

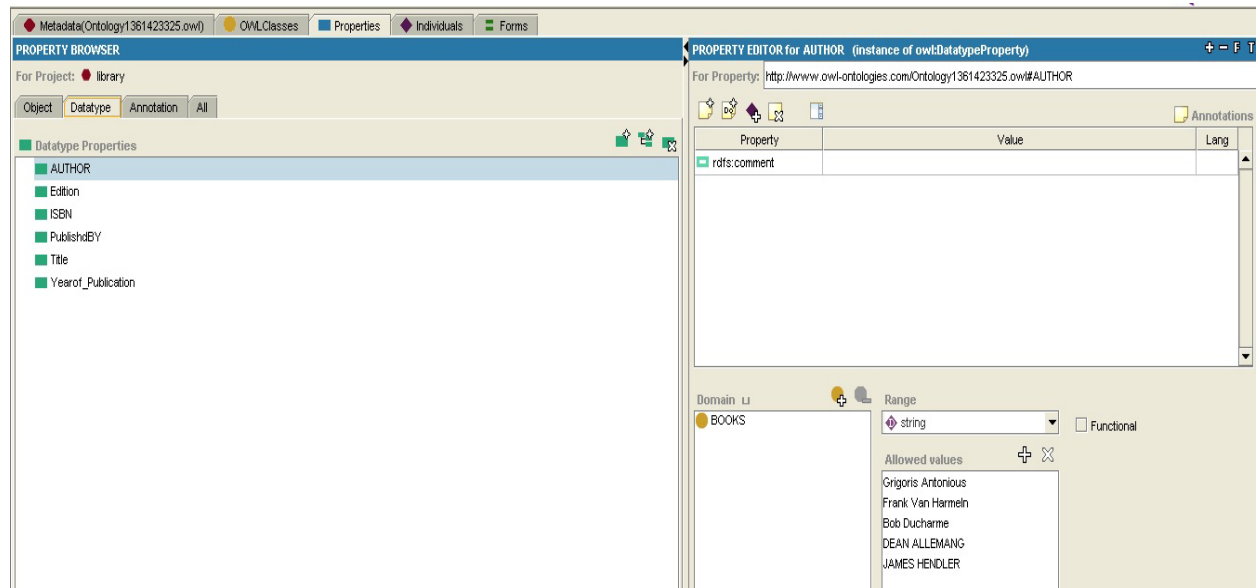


Fig. 9: Data type properties

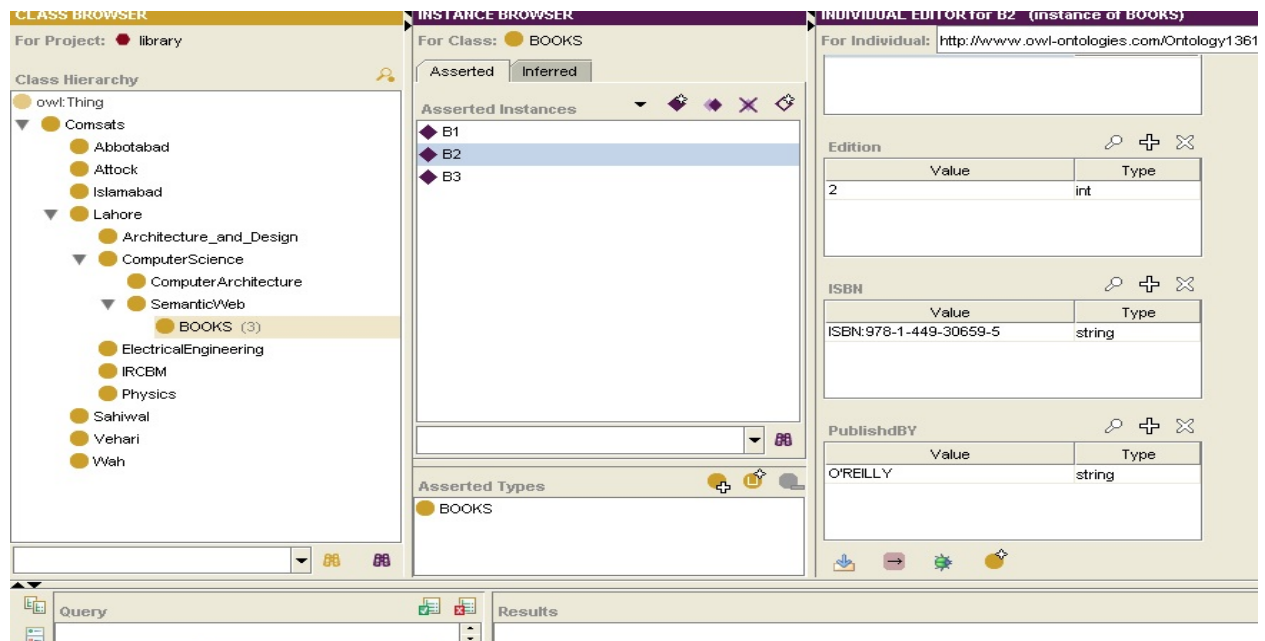


Fig. 10: Individuals properties

```
<owl:Class rdf:ID="Abbotabad">
  <rdfs:subClassOf rdf:resource="#Comsats"/>
</owl:Class>
```

Abbotabad was campus of Comsats. Comsats is the top hierarchy of the ontology model. Abbotabad is the subclass of `rdf:resource = "#Comsats"`. Subclass shows the interdependency between the different individuals. It just shows the generalization of relationship of all the classes. We can set the two properties as different. We said that the class Abbotabad and Attock both are different types of classes. Than in this case, there was

the setting of disjoint properties. This disjoint property helps the classes to differentiate with each other.

**Disjoint classes (Abbot bad: Attock):** Above statement tell us that Abbotabad and Attock both are campuses of Comsats. Both are the classes of Comsats. The purpose of disjoint properties was to make both classes different. Classes and Subclasses can relate with each other on the basis of instances. On the basis of instances we can develop relationship between the different classes and subclasses. We can develop and build the properties based on the properties like has of

and other properties just like Abbotabad and Attock has library. In this example has shows the object property.

**Object property assertion (haslibrary: Abbotbad: Attock):** We can build restriction on the classes by defining the domain and range of the classes. This range and domain shows the restriction and a limit on the classes beyond which the classes cannot cross. There is hierarchy tree between the classes and subclasses (Fig. 8).

The data type properties can set into the protégé for showing the relationship between the classes and sub classes. On the basis of data type properties we can develop the relationship and restriction between the two classes (Fig. 9).

We can also define the individuals of any class and can also set the properties of that individual. For example Books has three individuals. Individual has named as B1, B2 and B3. B2 has the properties like Title, Author, Year of Publication, Published By and ISBN number. We can define the value of each property related to the individuals (Fig. 10).

If we want to show the code of this ontology than by clicking on the RDF/XML source code, we can show the source code of the ontology. When the ontology is build, than SPARQL query is applied to get result. By applying the SPARQL query on the ontology model, we can extract the data. For some specific purpose applying query in the semantic web is the efficient way.

## DISCUSSION

With the help of ontology of any web application we can set the properties of that application with their specific properties and can also apply the limitation on these properties. Ontology is nothing it is just the relationship between the things and how these things are depend dent on one another. We proposed the ontology building method for the web application. Protégé Gómez-Pérez *et al.* (2004) is an open source tool to model a schema with their semantics (Liyang (2011), DuChame (2011), Toby *et al.* (2009)). Protégé can be used to load, edit and save ontologies in the bundle of formats like RDF, XML, UML, RDF PLUS and OWL etc.

## CONCLUSION

Keeping the whole structure of Library system, we can build the ontology of the Library System. Moreover, properties related to the ontology building are carefully identified. We must be careful about this. If there is some little mistake we cannot extract the data properly. The properties of ontology building are classes, subclasses, individuals, functional and transitive properties etc. We will use SPARQL for

querying the name space search engine. Java using a classified Jena APIs package to give support to SPARQL. User can generate their queries. Finally, query will be taken by user via user interface, which contain all the information about the books in library which the user want to search and mention in the query.

## RECOMMENDATIONS

Many researchers are working on the XML and RDF databases. On these databases, the ontology is build and after that the SPARQL query is applied. Our research team is working on the mapping and binding of xml and owl ontology for SPARQL. After binding of these ontologies, we can apply the SPARQL query on these ontologies. These researches are very helpful in our web application. It will help us to improve web applications.XML schemas for different web applications for search interfaces will be integrated to form meta search engine. For detailed ontology building, we include images/photographs of person in any web applications that is being imported from different image processing software as an input. We may build the library system of seven campuses by binding the ontology of different campuses and may integrate this large web application with the server and put many SPARQL queries.

## REFERENCES

- Aditya, K., P. Bijan and H. James, 2006. SMORE-Semantic Markup, Ontology and RDF Editor. Retrieved from: [www.dtic.mil/cgi-bin/GetTRDoc?AD=ADA447989](http://www.dtic.mil/cgi-bin/GetTRDoc?AD=ADA447989).
- Berners-Lee, T., J. Hendler and O. Lassila, 2001. Semantic web, a new form of web content that is meaningful to computers will unleash a revolution of new possibilities. *Sci. Am.*, 284(5): 34-43.
- Daconta, M.C., L.J. Obrst and K.T. Smith, 2009. *The Semantic Web: A Guide to the Future of XML, Web Services and Knowledge Management*. Wiley Publishing, Indianapolis.
- Dean, A. and H. Jim, 2011. Semantic Web for the working oncologist: Effective Modeling in RDFS and OWL. In: Denise, E.M. and P. Morgan (Eds.), *Kaufmann in Elsevier*.
- DuChame, B., 2011. *Learning SPARQL*. O'Reilly Media, pp: 258.
- Gómez-Pérez, A., M. Fernández-López and C. Oscar, 2004. *Ontological engineering: With examples from the areas of knowledge management, e-commerce and the semantic web*. 1st Edn., Springer-Verlag, New York, pp: 415.
- Grigoris, A. and F. van Harmelen, 2004. *A Semantic Web Primer*. MIT Press Cambridge, Massachusetts London, England.



- Herbert, S., 2002. Java 2: The Complete Reference. In: Brandon A. Nordin (Ed.), 5th Edn., McGraw-Hill, New York.
- James, B., Franc, B. Ois, F. Tim and S. Sebastian, 2008. In: Semantic Web Query Languages.
- John, H., F. Matthew, B. Ryan, A. Perez-Lopez and D. Mike, 2009. Semantic Web Programming. Wiley, John and Sons, Incorporated, ISBN: 9780470418017.
- Jonathan, H., 2004. In: A graph model for RDF: Diploma Thesis. Technische Universität Darmstadt Universidad de Chile.
- Kendall, G.C., F. Lee and T. Elias, 2008. In: SPARQL Protocol for RDF-- W3C Recommendation 15 January 2008.
- Li, D., P. Kolari, Z. Ding and S. Avancha, 2007. Using Ontologist in the Semantic Web: A Survey. TR CS-05-07, Maryland Baltimore County.
- Liyang, Y., 2011. A Developer's Guide to the Semantic Web. Springer, Heidelberg, New York.
- Malik, S.K., N. Prakash and S.A.M. Rizvi, 2010. Developing an university ontology in education domain using protégé for semantic web. Int. J. Eng. Sci. Technol., 2(9): 4673-4681.
- McBride, B., D. Boothby and C. Dollin, 2010. An Introduction to RDF and the Jena RDF API. Retrieved form: [http://jena.sourceforge.net/tutorial/RDF\\_API/inde.html](http://jena.sourceforge.net/tutorial/RDF_API/inde.html).
- Michael Grobe, SIGUCCS 2009. RDF, Jena. SparQL and the Semantic Web. pp: 131-138.
- Natalya, F.N. and D.L. McGuinness, 2009. Ontology development 101: A guide to creating your first ontology. Stanford Knowledge Systems Laboratory Technical Report KSL-01-05 and Stanford Medical Informatics Technical Report SMI-2001-0880.
- Paul, D., T. Valentina and I. Luigi, 2007. Ontology module Extraction for ontology reuse: An ontology engineering perspective. Proceeding of 16th ACM Conference on Information and Knowledge Management (CIKM), pp: 61-70.
- Rubbani, H.H., 2007. Semantic Web Solutions: Thesis report in IT University of Copenhagen. Retrieved from: [www.itu.dk/~hammad/thesis/Final%20Thesis.pdf](http://www.itu.dk/~hammad/thesis/Final%20Thesis.pdf).
- Sana, R., 2012. Semantic modeling of COMSATS university fee structure. M.S. Thesis, COMSATS Institute of Information Technology, Lahore, Pakistan.
- Semantic Web Architecture, Ontologies and Semantic web. Retrieved form: <http://www.obitko.com/tutorials/ontologies-semantic-web/semantic-web-architecture.html>.
- Smith, M.K., C. Welty and D. McGuinness, 2004. OWL Web Ontology Language Guide. W3C Recommendation 10 February 2004.
- Sven, G., 2011. Data Management and Query Processing in Semantic Web Databases. Berlin, Heidelberg, New York, pp: 1-270.
- The Future of the Web is Semantic, 2005. Ontologies form the backbone of a whole new way to understand online data" by developer works IBM. Retrieved from: [www.ibm.com/developerworks/library/wa-semweb/](http://www.ibm.com/developerworks/library/wa-semweb/).
- Toby, S., E. Colin and T. Jamie, 2009. Programming the Semantics Web.O' Reilly Media, Inc. 1005 Gravenstein Highway North, Sebastopol, CA 95472, pp: 84, ISBN: 978-0-596-15381-6.