A Study of Semantic Web and its Applications

COEN 380, Spring 2017
Group 1

Anita Patnaik
SCU ID - W1026651
Department of Computer Engineering
Santa Clara University,CA,USA
apatnaik@scu.edu

Vishaka Balasubramanian Sekar
SCU ID - W1362645
Department of Computer Engineering
Santa Clara University,CA,USA
vsekar@scu.edu

Mose Yoloye
SCU ID - W1018341
Department of Computer Engineering
Santa Clara University,CA,USA
myoloye@scu.edu

Aishwarya Bhide
SCU ID - W1261585
Department of Computer Engineering
Santa Clara University,CA,USA
abhide@scu.edu

,
,

***Abstract*—Semantic Web is a method of annotating the data on the web to create a metadata-rich Web of resources that can describe themselves not only by how they should be displayed Hypertext Markup Languages (HTML) or syntactically Extensible Markup Languages (XML), but also by the meaning of the metadata for machine- readability.**

In this paper we present an outline of the semantic web stack We also present a study on the various ontologies available to describe the WWW, along with their usage, and their associated data description languages. This paper also discusses about SPARQL, a query language used to query the RDF data.We present a comparison between RDF databases and other types of data models that exist. The applications of Semantic Web in real world applications are presented and the challenges in its implementation are outlined.

***Keywords*—Semantic web, RDF, RDFS,XML, SPARQL, semantic web application, Databases**

I. Introduction

Semantic Web is an extension of the World Wide Web with the enables us to write data-stores , vocabularies and rules for handling data. It helps in better organization and sharing of data amongst users and between machines. Artificial intelligence and machine learning are two key areas of development nowadays. Having a collection of data that is readable by both humans and machines is the main idea behind Semantic Web .

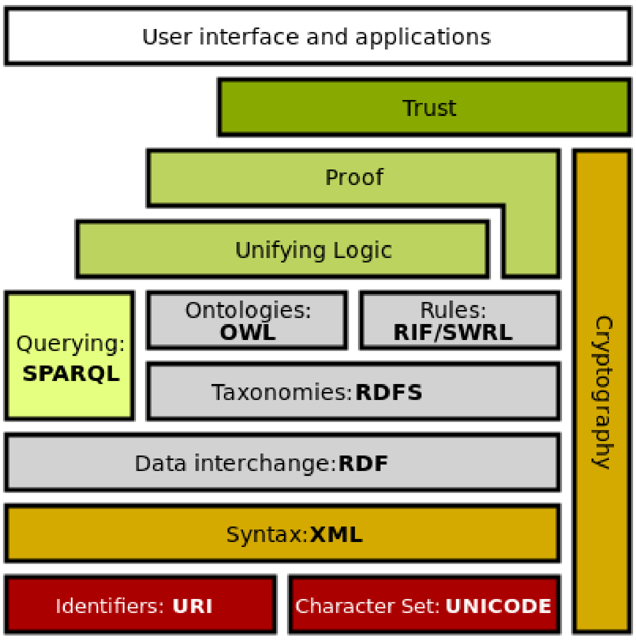

Fig. 1. Web Semantic Stack

Tim Berners Lee, who coined the term “Semantic Web” defined the it as "a web of data that can be processed directly and indirectly by machines". Which means, semantic web is a collection of data linked by hyperlinks in a human readable format and also having a meta-data about the pages and the interconnections among them to be processed by a machine. A standard format used across all documents makes this possible.

II. RDF AND OWL

A. Need for Semantic Web

The web documents on the web are created on Hyper Text Markup Language that models the layout of the content on the web page. The Cascading Style Sheets (CSS) enhance the visual representation of the data. However, this does not model the relationships between the content of the webpage accurately.

For instance, if we had to catalog the content on this paper as a web page , HTML would represent it as

```
<meta-name="keywords"
 content="computing, computer studies,
 semantic web" />
```

A more real-world representation in RDF would be

```
<rdf:subject= “Term project”>
<rdf:domain >COEN380</rdf:domain>
</rdf:subject>
```

The above simple annotation of the data on the web is simple, but ambiguous and inextensible .The Semantic Web takes the solution further. by combining the following technologies: Resource Description Framework(RDF), OWL(Web Ontology Language) , and XML.

B. RDF

Ontology is a description of data in a domain and the associated information is structured [1].Resource Description Framework (RDF) is the W3C standard model that is used for describing the resources on the Web . RDF was built on the the XML framework, and has the ability to merge the data even if the underlying structures are different.

Due to the large amount of information on the WWW, cataloguing that data is very difficult,.RDF is used to make the process automated by making the metadata understandable by all the clients and servers that exchange data .

Due to their homogeneous structure, RDF databases can be seen as labeled directed graphs.It describes resources that are identified by a URI , in terms of properties and their values.

Tim Berners Lee invented the WWW can be modeled as the below graph. The nodes representing strings are denoted by squares, the ovals represent the resource. The predicate is depicted in the relationship type.

Fig 3. :RDF

This is can be logically modeled as the the ER diagram, where the Entity is the resource and the Value is the property, and the arrow defines the relationship.

We need a serial , textual representation of the above semantic data. RDF supports many formats the popular among them being

- N Triples - easy to parse triples
- RDF-XML - Which is a XML form of rdf tags

In RDF the data can be represented in the form of triples <a P b>. Here, a is subject , P is the property and b is the value of that property.This triple is called the
< *Subject-predicate-object*>.

URI does not have to be an actual address on the Web, it can be any address of an entire website,or of a webpage , which returns a 300 -status message in the HTTP GET response can be treated as an URI.Each triple encodes the binary relation predicate between subject and object, i.e. represents a single knowledge fact. where each triple defines an edge from the subject to the object node under label predicate[2] as shown in Figure 2.

The properties or predicates that can be supported in RDF are :

- `type`
  `<COEN380> type <Course>`

- `subTypeOf`
  `<COEN380> subTypeOf<Database Course>`

- `subClassOf`
  `<COEN380> subClassOF<Masters level Course>`

- `domain, range`
  `<COEN380> domain<Databases>`

C . RDFS

Resource Description Framework Schema (RDFS) is a set of classes with certain properties using the RDF data model, providing basic elements for the description of ontologies, intended to structure RDF resources. These resources can be saved in a triplestore to reach them with the query language SPARQL.

It also extends definitions for some of the elements of RDF, for example it sets the domain and range of properties and relates the RDF classes and properties into taxonomies using the RDFS vocabulary.Classes are also resources, so they are identified by URIs and can be described using properties. The members of a class are instances of classes, which is stated using the rdf:type property.[3]

Element	Class of	rdfs:subClassOf	rdf:type
rdfs:Resource	all resources	rdfs:Resource	rdfs:Class
rdfs:Class	all classes	rdfs:Resource	rdfs:Class
rdfs:Literal	literal values	rdfs:Resource	rdfs:Class
rdfs:Datatype	datatypes	rdfs:Class	rdfs:Class
rdf:XMLLiteral	XML literal values	rdfs:Literal	rdfs:Datatype
rdf:Property	properties	rdfs:Resource	rdfs:Class
rdf:Statement	statements	rdfs:Resource	rdfs:Class
rdf:List	lists	rdfs:Resource	rdfs:Class
rdfs:Container	containers	rdfs:Resource	rdfs:Class
rdf:Bag	unordered containers	rdfs:Container	rdfs:Class
rdf:Seq	ordered containers	rdfs:Container	rdfs:Class
rdf:Alt	containers of alternatives	rdfs:Container	rdfs:Class
rdfs:ContainerMembershipProperty	rdf:_1... properties expressing membership	rdf:Property	rdfs:Class

Fig 3. :RDFS properties

D. OWL

Web ontology Language (WOL has been re arranged to form OWL for better readability).RDF is limited in enforcing the limitations whereas OWL has a richer vocabulary and stronger syntax than RDF.This results in OWL being is a stronger language with greater machine interpretability than RDF.

OWL consists of classes. A class is a group of objects .It has other classes,, instances of classes and restrictions for each property.Individuals in OWL are real world objects linked to other individuals through classes.The OWL abstract syntax presents an ontology as a sequence of *annotations*, *axioms* and *facts*. Annotations carry machine and human oriented meta-data . Information about the classes, properties and individuals that compose the ontology is contained in axioms and facts only[4]

```
<owl: Class rdf:about = "Tim Berners Lee">
```

OWL roles are similar to RDFS properties, they are of two types

- Data type properties

Datatype properties are relations between instances of classes and RDF literals or XML schema datatypes. For example, modelName (String datatype) is the property of Manufacturer class. They are formulated using owl:DatatypeProperty type.

- Object Properties

Object properties are relations between instances of two classes. For example, ownedBy may be an object type property of the Vehicle class and may have a range which is the class Person. They are formulated using owl:ObjectProperty.[4]

OWL is not like a database system no requirement that the only properties of an individual are those mentioned in a class. Also classes and properties can have multiple "definitions" statements about individuals need not be together in a document[5].

OWL is used define domain , range, hierarchies of the classes then define the them with properties and restrictions. THe restrictions can be

- `owl : domain`
- `owl: range`
- `owl : disjointWith`
- `owl: distinctMembers`
- `owl:AllDifferent`
- `owl:equivalentClasses`
- `owl:isPartOf`

Figure below shows an collection of classes belonging to a class 'Continent'[6].

```
<owl:Class rdf:about="Continent">
  <owl:oneOf rdf:parseType="Collection">
    <owl:Thing rdf:about="#Europe">
    <owl:Thing rdf:about="#Asia">
    <owl:Thing rdf:about="#Africa"/>
    <owl:Thing rdf:about="#NorthAmerica"/>
    <owl:Thing rdf:about="#SouthAmerica"/>
    <owl:Thing rdf:about="#Australia"/>
  </owl:oneOf>
</owl:Class>
```

Fig 4. :Owl

The following table presents a comparison between RDF, RDFS, and OWL [7]

TABLE I : Comparison between RDF, RDFS and OWL

Properties	**RDF**	**RDFS**	**OWL**
Class	Yes	Yes	Yes
Data	Yes	Yes	Yes
Object	Yes	Yes	Yes
Domain & Range	Yes	Yes	Yes
Annotation	No	Yes	Yes
Individual	Yes	Yes	Yes
Graph	RDF	RDFS	Onto
Inverse	No	Yes	Yes
Inference	Poor	Good	Better
Indexing	Poor	Good	Better
Searching	Poor	Good	Better
SPARQL Query	No	Yes	Yes
DL Query	No	Yes	Yes
Prefix	RDF	RDFS	Owl

III. SPARQL

The Semantic Web is an addition to the current web in which a well-defined meaning is given to the information to discover the knowledge in order to improve the human-computer cooperation. For such knowledge discovery, the goal was to build a description language with the help of traditional semantics so that the semantics can be processed with systems and can be made understandable to humans. W3C proposed a framework for representing information in the web called RDF (Resource Description Framework) which represents information as a graph, using a set individual objects and a set of connections[8]. This framework provides schema definition language (RDFS) to define new vocabulary. But, querying RDF is a challenge because URIs need to be dereferenced when an interconnection is present between RDF graphs[9]. Also, RDF graphs are naturally incomplete and with the new information getting in, adding that information is an important feature.[9] The new information which is getting added is the vocabulary which might have a predefined semantics which could create difficulty in knowledge discovery. To overcome these challenges, a query language for RDF, called SPARQL, was proposed by W3C as a query language for the distributed graphs.

SPARQL stands for SPARQL Protocol And RDF Query Language[10], which is a graph matching query language proposed by World Wide Web Consortium for querying RDF graphs. SPARQL uses Linked data model as its default This is a semantic query language for databases which is used to obtain and modify data stored in RDF standard format. SPARQL is one of the most important technologies of semantic web. It can provide results in various formats such as, XML, JSON, HTML, RDF, N-Triples and Turtle. SPARQL query format is divided in three parts; triple pattern, conjunction and disjunction. As explained in W3C recommended documentation of SPARQL [10][11], "Most forms of SPARQL queries contain a set of triple patterns called a basic graph pattern[10][11]. Triple patterns are like RDF triples except that each of the subject, predicate and object may be a variable [10][11]. A basic graph pattern matches a subgraph of the RDF data when RDF terms from that sub-graph may be substituted for the variables" [10][11]. SPARQL allows users to fire queries against "key-value data supported by RDF.[12] The whole database Is a set of "subject-predicate-object" Triples[12]. RDF data can also be considered to be SQL relational database[12] where the table contains subject column, predicate column and object column such that, all the triples are the separate rows with subject as a primary key[12], each possible predicate[12] to be a column and the object as a value from the heterogeneous object column. However, SPARQL becomes more powerful when works when the columns contain multiple values for a single key and column is joinable variable for the query[12]. For the data which has inherent schema in the data itself, SPARQL supports analytics queries such as sort, aggregations, join.

Users query RDF using SPARQL by specifying the template. These templates are compared with the graph components and when the data matches with these templates, the matched data is returned as the output of the query. For example, a query returns the managers from the sales_department of a company named, XYZ_company. This query execution involves graph pattern-matching. This query finds out the entities which are “managers” from the available data set, with two connections labeled as “name” which connects “company” to “XYZ_company” and another connection labeled as “department” connects “company” to “sales_department”. Here entities that match the pattern of the query can be referred with the variable name “?company(the ? character is used to indicate pattern variables)”[11]. As we are looking for the managers from the sales_department in XYZ_company, the entities must be linked to the entity “manager” in the data set with connection to other entity company labeled to be “manages” and all the entities matching the template are given out as the output.

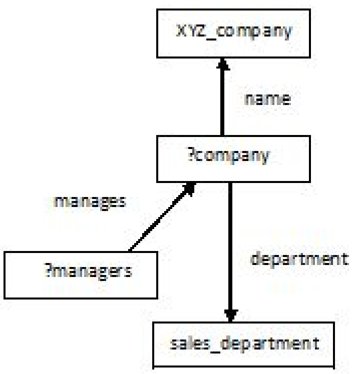

Fig 5. Example of SPARQL query

SELECT ?manager ?company
WHERE{
?manager manages ?company .
?company name XYZ_company .
?company department sales_department . }

SPARQL queries RDF graph database which are considered to have subject-predicate-object triples. Hence, triple templates are used while querying the RDF graph data. If RDF data is stored in relational “triples”[11], then SPARQL queries can be converted into the SQL queries[11] and the output can be found out by self joining triple tables for each SPARQL condition[11]. SPARQL plays a vital role in improving data integration in semantic web because it can work efficiently with the linked databases and graph databases to put together the linked information, which gets updated or added every second, to give out the precise output.

The latest W3C recommendation of SPARQL is SPARQL 1.1.[10] This version has new features which are aggregation for semantics. Also, a new operator SERVICE is added which distributes query execution over the data and results in faster performance[9]. The new feature avails property paths based on regular expressions[9].

We are living in the world of “Big Data”. The data is increasing which in all sorts of forms such as graph data, unstructured data, linked data etc. The linked data as well as distributed data are increasing in a huge amount and it seeks considerable attention throughout the research community. This technology and its potential to query linked as well as graph databases have inspired data science practitioners to search data over the years. As the research in distributed databases progressed, various approaches for managing data over SPARQL Endpoints are being introduced. SPARQL is the default data model of Linked Data deployment. SPARQL Endpoints access mechanisms are provided by many Linked Open Data(LOD) repositories[13]. There are some research is being done and projects are running to make dynamic SPARQL query writing easy for various applications[14]. The main goal is to make an autocomplete form which is reusable in different applications. This form must be created with the latest up-to-date ontologies. This makes the process of using Linked Open Data easily available to developers of the applications [14]. This is done using a server that returns a JSONP format through an API of the server for every SPARQL query sent by the application and this is how the autocomplete form works[14]. It is implemented with AngularJS and helps to write SPARQL keywords, ontologies and properties[14]. Such progress in SPARQL technology improves the query performance and improves data integration as well.

Another important area where the research is being done is the web of linked data. This web of linked data creates a single distributed open dataspace. Due to its openness, it is very difficult to keep track of the sources of the data coming from all over the world, but this could be relevant to the query to answer precisely[15]. The traditional research on query processing does not address this challenge and the new research is going on and new approach is proposed to precisely answer the SPARQL queries that are being fired on the web of linked data[15]. This new approach discovers the source of the data i.e. where the data is coming from, for answering the queries with more perfection. This is done by dereferencing URIs and by following RDF links[15]. The URIs are resolved over the HTTP protocol into RDF data which is continuously added to the queried dataset[15]. Hence, new algorithms and approaches are proposed by Olaf Hartig, Christian Bizer, and Johann-Christoph Freytag using iterator-based pipelining. A formalization of the pipelining approach is introduced to show that classical iterators may cause blocking due to the latency of HTTP requests[15]. This blocking can be avoided using iterator paradigm[15]

A constant growth is seen in the production of the web data all over the world. Two essential of components of the semantic web are RDF and SPARQL.[9] Database research community is interested in the research of SPARQL as the performance of semantic web is highly based on this query language.[9] Technologies like SPARQL autocomplete form are being researched to create a novel solution to improve semantic web and data integration.

IV. Comparison Between RDF and Other Database Types

RDF has a graph database structure to connect data and resources. This differs from two other popular database types: Relational databases, which are typically seen in SQL-based databases, and Hierarchical Databases, the structure that XML follows [16]. This section describes the different types of databases and how they compare.

A. SQL and Relational Databases

Relational databases use tables to describe the structure of the data. The tables describe the properties that a certain piece of data has. Relations between different tables, and therefore between the data are accomplished using primary and foreign keys. Primary keys identify each record within a table. Typically, this is an ID number, but can also be any unique identifier for the record [17]. A foreign key relates one table to another table. For example, to describe college courses, there may be a table for the course and its information, and a separate table for the professor who teaches it. Each course would have a foreign key property denoting the primary key of its professor.

Relational databases are good for data that has a very specific structure. Each record in a relational database will generally have all properties defined that are specified in the table. Relational databases start to become less effective when some records have a certain property but other records do not and are filled with NULL values instead [18].

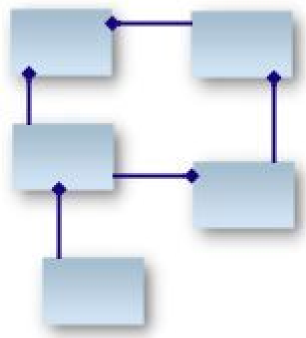

Fig 6. General structure of a Relational Database

B. XML and Hierarchical Databases

Hierarchical databases structure the data in parent-child relationships. The best example of this is XML. XML has one top-level node describing the general document. Within that top-level node, there are several child nodes describing the properties of the parent [16]. This hierarchical structure can go several levels deep. For example, to describe the courses that a department in a university offers, the top-level node could describe the department, with child nodes to describe the department's information. Then, another child node denoting the list of courses would contain each course and its information. The XML for this example is shown below:

```
<department>
   <name>Computer Engineering</name>
   <chair>John Smith</chair>
   <course_list>
      <course>
         <title>Databases</title>
         <number>280</number>
      </course>
      <course>
         <title>Data Mining</title>
         <number>281</number>
         <prereq>Linear Algebra</prereq>
      </course>
   </course_list>
</Department>
```

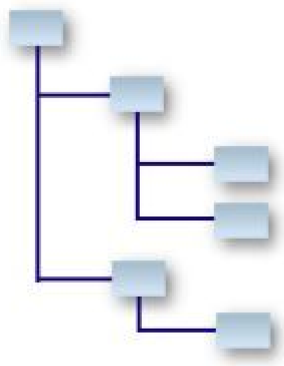

Fig 7. General structure of a Hierarchical Database

Hierarchical databases can add a bit more flexibility to data than what relational databases allow, as it is easier to denote child nodes of a parent as optional [19].

C. RDF and Graph Databases

Graph databases connect pieces of data to other data. The relations are arbitrary [16]. As stated earlier in this paper, RDF is based off of XML. However, RDF and its accompanying schemas allow for additional semantic information to be added to data, where XML cannot. Therefore, information can also be inferred from the current data. RDF allows for the flexibility to add new triples easily, extend the data, and create new relations [20].

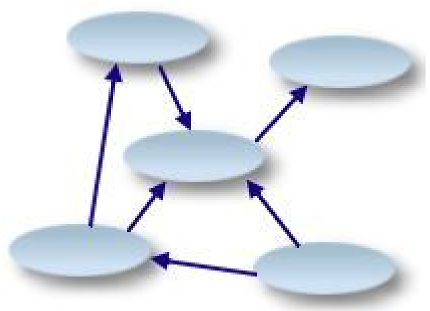

Fig 8. General structure of a Graph Database

As the whole Internet is essentially a graph of documents connected to each other, it makes sense to use a graph structure for the data as well [21]. Unlike relational databases, the graph database structure allows for more flexibility in adding relations to other resources. While XML is a big improvement from that, there are still issues with adding new relations. If using a schema, the whole schema would need to change, and furthermore, there are several ways to represent the same data in XML. RDF and ontologies provide great flexibility while still maintaining structure, and the form of the triples is both human and machine friendly.

V. Applications

In this section,we study the application of web semantics in the fields of education and information retrieval.

A. Semantic Web based Education System

The value of education in the society is well known. A nation's prosperity and economic growth depends on it. It is the key to the growth of human knowledge and open-mindedness. In this section we see how web semantics enhances the education system by providing education anywhere, anytime and to anybody. It provides a personalized learning system to the users, which is based on the concept of ontology. Thereby making it more relevant.

The traditional education system suffers from certain disadvantages that Semantic Web based Educational System tries to address.

- For example, currently, when searching on a topic of interest a lot of time is spent going through the links returned by the search engine and determining the relevant ones. Semantic Web focuses on returning a multimedia report by comparing and intelligently aggregating the search results.
- A lot of information is stored in various formats like web pages, blog discussions, videos etc. on various devices. Web Semantics aims at consolidating this data for the end user. Thus, allowing more time to think and work on the research topic rather than culling out the right links.
- At present the learning systems are geared towards presenting the same data to every learner. However the Semantic Web will enable information to be personalized based on the user profile.

Building a Semantic Web based Education System needs a combination of the following technologies:

- UML, to represent the flow between complex systems involved.
- XML, to exchange user defined information between systems.
- RDF and RDF-Schema, to represent objects and the relationships between them.
- OWL Ontology, to define knowledge of things and the relationship between them.
- Software, to interact with the various systems and

end users.

This learning system is composed of three actors – the learners, the teachers and the authors. The learners interact with the system and in order to receive educational content. Authors create the education content. The teacher's role is to present the content in appropriate order, monitor the student's progress and change direction/difficulty level of the course as appropriate. The software/teaching agent's job is to collect data from different sources, assimilate it and display it to the learner. As the course progresses the system develops a user model for the learner and tailors the learning material based on it.

Ontology forms the backbone of the Semantic web based education system. Ontology is the formal definition of terms of a domain and the relationship between these terms. Ontologies in education system are needed for building the domain and instructional knowledge. Building ontology is an iterative process. First the scope or the domain is determined by accessing who are the users, what kind of problem/questions are being addressed etc. Second, the terms that need to be formally explained are identified. A class hierarchy is established. Third, the maintenance process is determined. Finally, the ontology is evaluated on how well does it satisfy the user requirement. Building Ontology is an iterative process. These steps are repeated until the final ontology is obtained.

There are two main ontologies that need to be developed for a education based system - subject ontology and learner's ontology. The subject ontology acts as a classification system for the knowledge base. Any search result can be returned based on this classification. The learner's ontology deals with the teaching technique and sequence of learning. The results are displayed based on user's profile.

Ontology stores the metadata along with the data. To query the ontologies we use descriptive logic along with inference rule. An ontology is formed by developing a corpus from domain based texts/documents. Each of these areas are classified to have one or more knowledge points. When a user, using the learning system, raises a query, first the already existing question base is search and the results and concepts obtained are compared to the concepts from the question. If no match is found, the knowledge points from the corpus are searched and appropriate result is returned.
Below is a basic, architecture of a semantic web based education system

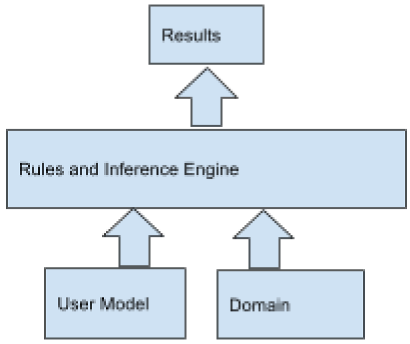

Fig 9 .Architecture for a web semantic based education system.

Ontologies are generated for the roles of author, teacher and learners. The Database and the world wide web act as the knowledge base. The Rule and Inference engine returns results based on the inputs from the user models and domain. For example, a user model for a student can be design. The results can be returned to the user, based on the match from the knowledge base and tailored to the student model. RDS, OWL and Web Mining can be used to design such a system. [22][23][24]

B. Information Retrieval and Semantic Web

Information Retrieval plays a key role in success of the web today. In this section we see how web semantics impacts the information retrieval on the web.

Current information retrieval systems on web is based on text based search, building text based indexes, keyword and thesaurus based matching. Web Semantic on the other hand deals with semantic web documents or text documents that are enhanced by semantic web annotations. These annotations makes the document metadata enriched and machine interpretable. We look at three approaches to semantic web search. In the first approach, the search remains similar to the present one with the inclusion of web semantic annotations to the text documents. In the second approach, the text data and the web semantic information is kept separately and reference each other. In the third approach, an independent crawler and an indexer is built for the semantically enhanced documents.

One of the major challenges of integrating the the semantically enhanced documents with the current search engines is that the search engines are based on term statistics and are not designed to process these mark ups. Although the RDF and OWL are termed as semantic markup languages. There are language representations and cannot be directly related to the text. Text representation, in turn, is difficult to convert and interpret for the web semantics inference engine. There is a lack of defined standards available to bind HTML and semantic markup together. The goal for building an Information Retrieval system, based on both principles, would be to tightly couple both search and inference engine. The system should support both text and semantic markups.

Considering only retrieval using semantically marked documents, the query received is also semantically marked. Since using the traditional search engine, the query is converted to text format. This process of encoding the semantic markup as text is known as *swangling*. The query is now submitted to the search engine. The results returned are scraped for semantic markups. The number of documents scraped depends on the requirements of the inference engine. These scrapped results are then filter based on relevance, authenticity and redundancy. This can be used as a feedback for the inference engine to retrieve better results.

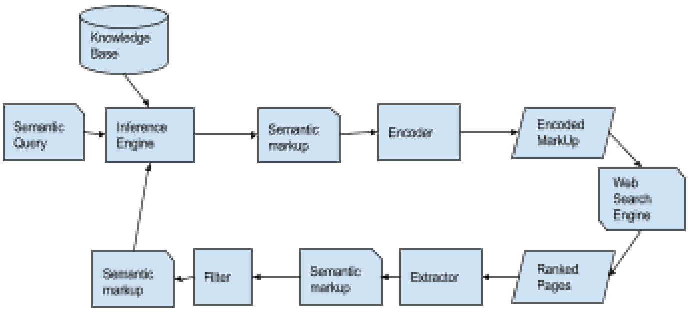

Fig 10 .Architecture of information retrieval system

This concept can be extended to include text in the query, in which case both text and 'swangled markup' is used as the input to the web search engine. The results returned, in this case, contain both text and semantic markup. Both are filtered to generate results. The filtered results forms the feedback for the inference engine to improve results.

We study three approaches, in this paper, to implement such an engine. The first prototype is OWLIR (Ontology Web Language and Retrieval). It uses text documents as well as semantic web pages. It can process both text and semantic markup queries. The text is annotated with semantic markups. The custom inference engine processes the metadata from the text to define relations and decides the scope of search.

The second prototype is Swangler. One of the major drawbacks in today's system is there are no standards in place for HTML document to refer or embed RDF in them. Also when RDF is included in the documents, the existing search engine treats it like plain text and applies tokenization rules of natural languages to it. Thus making it ineffective. The Swangler enhances the existing RDF documents with additional statements containing swangle terms. These terms can be picked by the traditional search engines and processed to return these documents when a match is found.

The third prototype is Swoogle. it is a crawler based Information Retrieval system. It provides search for semantic web documents. It is based on the fact the documents being dealt with have RDF content and are for machine processing. The metadata for these documents is stored and maintained. When queried, the search results returned are based on ranking generated by an algorithm similar to PageRank. The SWD rank also ranks the items based on document importance.

Factors that influence the process selected for adapting the information retrieval system to semantic web are - the kind of documents being processed, how the markup is being processed, the query size, how the tokenization is done, which triples are swangled etc. In conclusion, the information retrieval on semantic web deals with both semantic markup data and text embedded with semantic markup. Both inference and search engines are coupled together to provide results. [25][26]

VI. Challenges

Web Semantic promises better data integration and data reuse. It adds meaning to the content by developing machine interpretable documents that can be processed to give humans a better experience. However, it faces the following challenges:

- *The vastness of the web* - The automated reasoning system will have to deal with a huge amount of data. Each ontology in itself is huge and creating and maintaining is a difficult task.
- *The vagueness of term* - Terms like hot, cold, short, tall aren't quantified and their concepts may vary.
- *Uncertainty* - Conditions for a match are fulfilled for various categories but with different probabilities
- *Inconsistency* - When two or more ontologies are

combined and they have differences, it results in inconsistencies.

- *Deceit* - When the source of information is false. This results in credibility issues
- *Multiple data formats*- Two formats for viewing data, the human readable one and the machine interpretable one.
- *Dissimilarity to human languages*- It is not similar to a natural language and skills need to be learned in order to develop it.

VII. Conclusion

Web Semantic is a work in progress. It builds upon the idea of the web today and enhances it with better data integration, data detection and reuse. We described how data is represented and queried using web semantic technologies. Web ontology languages are extensible, flexible and powerful way of representing data on the web in terms of more meaningful metadata .Web semantics finds its application in various fields. In this paper, we looked at its application in the fields of education and information retrieval. Its application in the field of education can help achieve personalized education experience for all and will help students focus more time on processing and researching the material rather than searching for it. With inter-institutional knowledge sharing agreement and web semantics, we can achieve a standard education across institutions. Web Semantics also finds its application in the field of Information Retrieval. It faces a huge challenge in terms of integration with the current technology. We studied a few protocols through which we can achieve this integration between the inference engine and the existing search engines. Despite the challenges faced for web semantics implementation,the fundamental advantage of a Semantic Web is that we can increase data generation, access, flow, integration, and comprehension using the very same open standards that drive the World Wide Web (URIs, HTTP, and HTML). This is the best way to achieve useful AI, based existing Web infrastructure.

References

[1] B. Kapoor and S. Sharma, "A Comparative Study of Ontology building Tools in Semantic Web Applications", *International journal of Web & Semantic Technology*, vol. 1, no. 3, pp. 1-13, 2010.

[2] N. Kumar and S. Kumar, "Querying RDF and OWL data source using SPARQL", in *Internal Conference on Computing, Communications and networking Technologies (ICCCNT)*, 2013.

[3] "RDF Schema RDFS - Introduction to ontologies and semantic web - tutorial", *Obitko.com*, 2017. [Online]. [Accessed: 05- Jun- 2017].

[4] E. Babayants, "Information management services-Galaxy Consulting", *Galaxy Consulting*, 2017. [Online]. [Accessed: 05- Jun- 2017].

[5] P. Patel-Schneider, *An Example OWL Ontology*, 1st ed. 2004.

[6] S. Linckels, "Semantic Web - OWL", *Slideshare.net*, 2014. [Online]. [Accessed: 05- Jun- 2017].

[7] J. Kahn and S. Kumar, "Deep Analysis for Development of RDF, RDFS and OWL Ontologies with Protege", in *International Conference on Reliability, Infocom Technologies and Optimization (ICRITO)*, 2014.

[8] M. Grobe, "RDF, Jena, SparQL and the 'Semantic Web'", *Proceedings of the ACM SIGUCCS fall conference on User services conference - SIGUCCS '09*, 2009.

[9] M. Arenas and J. Perez, "Querying Semantic Web Data with SPARQL: State of the Art and Research Perspectives".

[10] "SPARQL Query Language for RDF", *W3.org*, 2008. [Online]. [Accessed: 05- Jun- 2017].

[11] J. Huang, D. Abadi and K. Ren, "Scalable SPARQL Querying of Large RDF Graphs".

[12] "SPARQL", *En.wikipedia.org*. [Online]. [Accessed: 05- Jun- 2017].

[13] N. Rakhmawati, J. Umbrich, M. Karnstedt, A. Hasnain and M. Hausenblas, "Querying over Federated SPARQL Endpoints ---A State of the Art Survey", *Arxiv.org*, 2017. [Online]. [Accessed: 05- Jun- 2017].

[14] A. Andreevski, R. Stojanov, M. Jovanovik and D. Trajanov, "Semantic Web Integration with SPARQL Autocomplete", in *12th International Conference on Informatics and Information Technologies*, 2015.

[15] O. Hartig, C. Bizer and J. Freytag, "Executing SPARQL Queries over the Web of Linked Data", *Lecture Notes in*

Computer Science, pp. 293-309, 2009.

[16] "Tutorial 1: Introducing Graph Data", *Linkeddatatools.com*, 2017. [Online]. [Accessed: 05- Jun- 2017].

[17] "A Relational Database Overview (The Java™ Tutorials > JDBC(TM) Database Access > JDBC Introduction)", *Docs.oracle.com*, 2017. [Online]. [Accessed: 05- Jun- 2017].

[18] C. Vicknair, M. Macias, Z. Zhao, X. Nan, Y. Chen and D. Wilkins, "A comparison of a graph database and a relational database", *Proceedings of the 48th Annual Southeast Regional Conference on - ACM SE '10*, 2010.

[19] "Guide to the W3C XML Specification (XMLspec) DTD, Version2.1", *W3.org*. [Online]. [Accessed: 05- Jun- 2017].

[20] J. Sequeda, "Introduction to: RDF vs XML - DATAVERSITY", *DATAVERSITY*, 2017. [Online]. [Accessed: 05- Jun- 2017].

[21] "Introduction to the Semantic Web", *Cambridge Semantics*. [Online]. [Accessed: 05- Jun- 2017].

[22] L. M Mathews and B. R K Rao, "An Overview: Semantic Web based Education", *International Journal of Computer Applications*, vol. 26, no. 2, pp. 18-22, 2011.

[23] J. Ohler, "The Semantic Web in Education", *Educause Quarterly*, vol. 31, no. 4, pp. 7-9, 2008.

[24] M. Buffa, S. Dehors, C. Faron-Zucker and P. Sander, "Towards a Corporate Semantic Web Approach in Designing Learning Systems. Review of the Trial Solution Project", 2005.

[25] N. Noy and D. McGuinness, "What is an ontology and why we need it", *Protege.stanford.edu*. [Online]. [Accessed: 05- Jun- 2017].

[26] T. Finin, J. Mayfield, A. Joshi, R. Cost and C. Fink, "Information Retrieval and the Semantic Web", 2005.