Information Science and Technology
ISSN: 0976-917X & ISSN: 0976-9188, Volume 2, Issue 1, 2012, pp.-31-35.
Available online at http://www.bioinfo.in/contents.php?id=77

ONTOLOGICAL WEB-SEMANTIC METHODOLOGY FOR E-LEARNING

VEKHANDE P.* AND JANWE

Department of CSE, Rajiv Gandhi College of Engineering, Research & Technology, Chandrapur, MS, India.
*Corresponding Author: Email- amolmandar@gmail.com

Received: February 21, 2012; Accepted: March 15, 2012

Abstract- The Semantic Web is simply a web of data described and linked in ways to establish context or semantics that adhere to defined grammar and language constructs. With the help of standard control structures in the programming languages, we can achieve this and also with many other programming techniques such as database querying. This nonstandard, dispersed way of programmatic semantic places restrictions on it and makes it unnecessarily complex, essentially obfuscated. Standing alone, the meaning of various terms such as Monument is simply lost. The Semantic Web addresses semantics through standardized connections to related information. This includes labeling data unique and addressable. Thus, your program can easily tell if this Monument is the same as another Construction Site in reference. Each unique data element then connects to a larger context, or web. We can use Object oriented principles like Inheritance, Aggregation, and Composition etc. to obtain Web of knowledge. That will enable us to use popular languages like Java and C#.Net to make intelligent web. We in this paper will try to explore these features.
Keywords- Semantic web, Java, C#Net.

Citation: Vekhande P. and Janwe (2012) Ontological Web-Semantic Methodology for e-Learning. Information Science and Technology, ISSN: 0976-917X & ISSN: 0976-9188, Volume 2, Issue 1, pp.-31-35.

Introduction

In the current status of the Ontological Semantic we see it as the extension of the current web. In the current web we provide extra information in the form of Meta data which is tagged with every document. This meta-information defines what the information (documents) is about in a machine processable way. The explicit representation of meta-information, accompanied by domain theories, will enable a web that provides a qualitatively new level of service [1].

It will weave together an incredibly large network of human knowledge and will complement it with machine processability.

Ontologies offer a way to cope with heterogeneous representations of web resources. The domain model implicit in an ontology can be taken as a unifying structure for giving information a common representation and semantics. The applications can choose to standardize on specific Ontologies and translate to ones employed by other applications. Advanced Semantic Web applications could automatically align vocabularies using advanced information techniques that logically employ the many paths within the Semantic Web. Thus, the rich relationships and the many relationship types each contribute to establish semantics the Semantic Web.

Keywords and Semantic Words

In the general use of a word we normally do not associate meanings with the word. A variable in a programming language can hold values but not the meaning associated with that. The semantic words will have some meaning associated with them.

Semantics give a keyword symbol useful meaning through the establishment of relationships. For example, a standalone keyword such as Mobile exists on a web page devoted to ontologies. The <META> tag surrounds the *Mobile* keyword to indicate its importance.

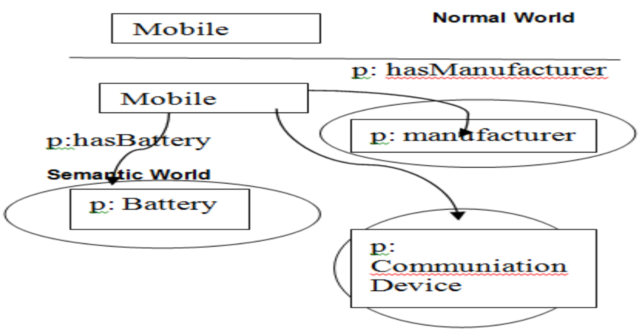

Fig. 1- Ontological World

However, does mobile mean transportation ontology or Ontologies that focus on cellular devices? The awkwardness of the previous sentence points out the difficulty in simply expressing semantics in English. Semantics are left for the human reader to interpret. However, if the keyword relates to other keywords in defined relationships, a web of data or context forms that reveals semantics. So Mobile relates to various other keywords such as Affording change, transportation, cell phone, and so on-the relationships expose semantics.

The Semantic Web is simply a web of data described and linked in ways to establish context or semantics that adhere to defined grammar and language constructs [1, 2].

Comparison of WWW and SW

Attributes	WWW	SW
Fundamental component	Unstructured content	Formal statements
Primary audience	Humans	Applications
Links	Indicate location	Indicate location and meaning
Primary vocabulary	Formatting instructions	Semantics and logic
Logic	Informal/nonstandard	Description logic

Semantic Web Complexity

Semantic Web statements employ a Semantic Web vocabulary and language to identify the different types of statements and relationships. Various tools and application frameworks use the statements through an interpretation of the vocabulary and language. Exploring and. applying these tools and frameworks in relationship with the Semantic Web keywords is the focus of this paper. The Semantic Web offers several languages. Rather than have one language fit all information and programming needs, the Semantic Web offers a range from basic to complex. This provides Semantic Web applications with choices to balance their needs for performance, integration, and expressiveness.

A set of statements that contribute to the Semantic Web exists primarily in two forms;

- Knowledgebase
- Files.

Feature	RDBMS	Knowledgebase
Structure	Schema	Ontology statements
Data	Rows	Instance statements
Administration language	DDL	Ontology statements
Query language	SQL	SPARQL
Relationships	Foreign keys	Multidimensional
Logic	External of database/triggers	Formal logic statements
Uniqueness	Key for table	URI

Knowledgebase offer dynamic, extensible storage similar to relational databases. A short comparison will make it clearer.

Components of Semantic Web

Semantic Web programming consists of core components: statements, the URI, an ontology, and instance data managed and formed through the various construction tools, interrogation tools, reasoners, and rules. The Semantic Web is solidly grounded on graph theory and description logic. It provides a knowledge representation that is defendable and worthy of our investment. There are two primary aspects to Semantic Web programming:

Knowledge representation

Application integration.

All this can be explained by the following diagram [2].

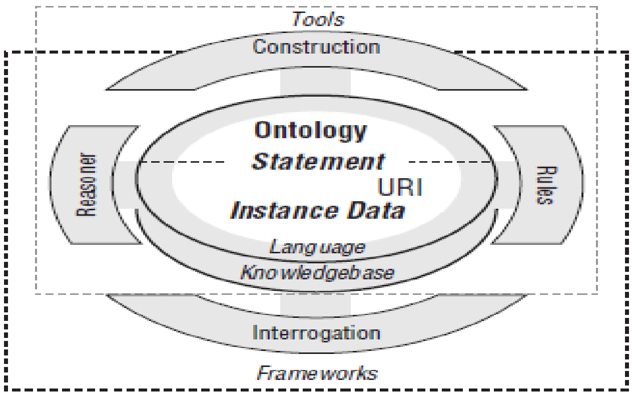

Fig. 2- Semantic Web

The core of the Semantic Web has Statement, URI, Ontology, and Instance Data.

- **Statement-** The statement forms the foundation of the Semantic Web. Each statement consists of multiple elements that typically form a triple. The triple consists of a subject, predicate, and object (e.g., Pen is Type Writing Tool).
- **URI-** A Uniform Resource Identifier provides a unique name for items contained in a statement across the entire Internet. Thus, each component of a statement-subject, predicate, and object-contains a URI to affirm its identity throughout the entire WWW.
- **Ontology-** Ontology consists of statements that define concepts, relationships, and constraints. Ontologies define a formal semantics for information allowing information processing by a computer. Ontologies define a real-world semantics allowing to link machine processable content with meaning for humans based on consensual terminologies. Sometimes, ontologies are confused with knowledge bases, in particular because the same languages (OWL, RDF-S, WSML, etc.) and the same tools and infrastructure can be used both for creating Ontologies and for creating knowledge bases. There is, however, a clear distinction: Ontologies are the vocabulary and the formal specification of the vocabulary only, which can be used for expressing a knowledge base. It should be stressed that one initial motivation for Ontologies was achieving interoperability between multiple knowledge bases. So, in practice, an ontology may specify the concepts “man” and

"woman" and express that both are mutually exclusive-but the individuals Amol, Mandar, and Sonia are normally not part of the ontology. Consequently, not every OWL file is an ontology, since OWL files can also be used for representing a knowledge base. There exist several approaches of classifying types of ontologies, namely by Lassila and McGuinness (Lassila & McGuinness, 2001) and by Oberle (Oberle, 2006, pp. 43-47) [4]. Lassila and McGuinness did order ontologies by increasing degree of formal semantics, while Oberle introduced the idea of combining multiple dimensions.

Semantic Web Application

To have an application that takes full advantage of the Semantic Web and its tools, applications must adapt to its expectations and impacts. We organize the programming impacts into four categories.

- **Web data-centric**

Semantic Web application should place data at its center. Data is key.

- **Semantic data**

Semantic Web application should place meaning directly within the data rather than within programming instructions or pushed out for user interpretation.

- **Data integration/sharing**

Semantic Web application should attempt to access and share rich information resources throughout the WWW when appropriate, including taking advantage of the many existing data sources.

- **Dynamic data**

Semantic Web application should enable dynamic, run-time changes to the structure and contents of your information. These four impacts potentially change the way you design and program an application. They guide your solution to make optimal use of the Semantic Web. The Semantic Web enables higher levels of information expressiveness. Limits on information expressiveness challenge programming solutions. Variables, structures, relational tables, and so on all have their limits and peculiarities. Relationships take on a primary role in the Semantic Web. In fact, they are the very fabric of the Semantic Web. Object-oriented solutions make relationships secondary to the objects themselves. Relationships do not exist outside of an object. Relationships are dependent on their associated object class. Relationships cannot be repurposed for other classes. Relationships in the Semantic Web exist distinct from the concepts that they join. Relationships are free to join any collection of statements. This allows relationships to have inheritance and restriction rules of their own.

The Semantic Web offers flexibility with instances. An instance is not permanently bound to any class or set of classes. In fact, an instance can have no class at all and merely stand alone as an instance statement or be associated with multiple classes. This allows the application to add instances before it understands their connections to classes. Application can dynamically change the association of an instance with its class. It can also assign multiple classes to a given instance. This allows the flexibility to form and capture information independent from class definitions. These assignments of instances to class can occur at any time. Because our data is represented in a flexible model, it is easy for someone else to integrate information about espresso machine locations, allowing our application to cover not only restaurants and bars, but also coffee shops, book stores, and gas stations. A well-designed application should be able to seamlessly integrate new semantic data, and semantic datasets should be able to work with a wide variety of applications. It all depends on how well you manage to get your Ontology and establish proper relationship on to it. It can be seen that the language like java that supports full Object oriented programming, can be used effectively to integrate the web with semantic web concept. The Ontology that is defined on the one layer can be dynamically linked with other layer and fetch the meaningful information from it. This very technique can also be used to make out the contradictions in the Ontology definition. Let's assume that a public site holds an important information regarding people in a rdf format.

```
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix admin: <http://webns.net/mvcb/> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix swp2: <http://semwebprogramming.org/2009/ont/chp2#>.
< swp2:me>
rdf:type foaf:Person ;
foaf:depiction <http://semwebprogramming.org/semweb.jpg> ;
foaf:family name "Web" ;
foaf:givenname "Semantic" ;
foaf:homepage <http://semwebprogramming.org> ;
foaf:knows < swp2:Reasoner> , < swp2:Statement> , < swp2:
Ontology> ;
foaf:name "Semantic Web" ;
foaf:nick "Indra Jal" ;
foaf:phone <tel:91-0712-2243257> ;
foaf:schoolHomepage <http://www.web.edu> ;
foaf:title "Dr" ;
foaf:workInfoHomepage
<http://semwebprogramming.com/dataweb.html> ;
foaf:workplaceHomepage
<http://semwebprogramming.com> .
< swp2:Reasoner>
rdf:type foaf:Person ;
rdfs:seeAlso <http://reasoner.com> ;
foaf:mbox <mailto:amolmandar@hotmail.com> ;
foaf:name "Sulochana" .
< swp2:Statement>
rdf:type foaf:Person ;
rdfs:seeAlso <http://statement.com> ;
foaf:mbox <mailto:amolmandar@yahoo.com> ;
foaf:name "R.R.M Roy" .
< swp2:Ontology>
rdf:type foaf:Person ;
rdfs:seeAlso <http://ont.com> ;
foaf:mbox <mailto:amolmandar@gmail.com> ;
foaf:name "Raju Deshmukh" .
```

Number of copies: 1 copy for each issue

This collection of statements does not constitute ontology, although it refers to the various Ontologies or vocabularies in the prefix portion at the top. Ontology has information regarding classes and their relationships. This data only refers to potential Ontologies elsewhere.

Reasoner, Framework and Object Oriented Language

Now since we have obtained the basic structure for the ontology and we have as well gone through the semantic behind this, we can now try to ease out the association of the Meta part of the data with the data. This is done through some grammar understanding structure. Best way to achieve this is to get into the patterns and form a simple framework. This frame work will be responsible for having the rules associated with the each component of the data. Once we form the frame work then it just becomes the matter of making the classes and hence Objects from it and associate the instances of the classes with various Ontologies using the framework [2]. This concept will allow us to use the Ontology with the Object Oriented languages.

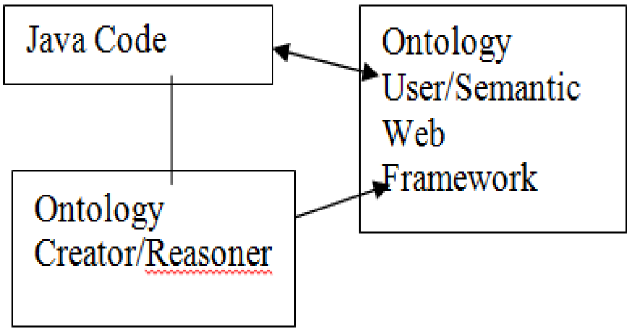

Fig. 3- Ontology & OOPs

```
public class PersonalSemanticWeb {
static String defaultNameSpace = "
http://semwebprogramming.org/2009/ont/cp2:#";
private Model friends = null;
public static void main(String[] args) throws IOException {
PersonalSemanticWeb psw = new PersonalSemanticWeb ();
//Load friends
System.out.println("Load Friends");
psw.populateFriends();
//...
}
// ...
private void populateFriends(){
friends = ModelFactory.createOntologyModel();
InputStream inFoafInstance =
FileManager.get().open("Ontologies/Friends.rdf");
friends.read(inFoafInstance,defaultNameSpace);
inFoafInstace.close();
}
```

Having created the required struture we can now exploit the real potential of Semantic webbing. Possibly, we have now utilized this graph in most heterogeneous manner. We can now get on to the real instance of the Ontology that was not directly related to the one which we have created. That will make the actual use of Semantic logics [5].

```
public static void main(String[] args) {
// ...
System.out.println("\nExploit Heterogenous Webbing");
pws.eHW(pws.friends);
// ...
}
// ...
private void eHW(Model model){
// focused search
runQuery (" select DISTINCT ?name where{ swp2:me
foaf:name ?name }", model);
}
private void runQuery(String queryRequest, Model model){
StringBuffer queryStr = new StringBuffer();
// Establish Prefixs
queryStr.append("PREFIX swp2" + ": <" + defaultNameSpace +
"> ");
queryStr.append("PREFIX foaf" + ": <" +
"http://xmlns.com/foaf/0.1/" + "> ");

//Now add query
queryStr.append(queryRequest);
Query query = QueryFactory.create(queryStr.toString());
QueryExecution qexec = QueryExecutionFactory.create(query,
model);
//Run Select
try {
ResultSet response = qexec.execSelect();
while( response.hasNext()){
QuerySolution soln = response.nextSolution();
RDFNode name = soln.get("?name");
if( name != null ){
System.out.println( "This is Cross Breeding " + name.toString() );
}
else
System.out.println("Not Possible Now!");
}
} finally { qexec.close();}
}
```

Conclusion

We have seen that the use of Semantic Web can be extended to the proper manipulations for e-learning in the popular Object Oriented languages like Java. We have seen that the Semantic Web if used properly can dynamically establish the relations among the Ontologies and its corresponding instances. In object Oriented Systems an instance has one to one relationship with its class, but in the Semantic Web we can cross this restriction by using the Object Oriented language only. An Ontology once created can get a Graph and thus can be utilized by other ways to manage that Graph such as e-learning. We loaded ontology and instance data, queried it, added more instance data, bound the instances to ontologies, and then aligned the ontologies. We used a reasoner to infer information into models based on the ontologies, and finally we went the other way and restricted the hellos to a subset of the initial data. First, we restricted it to friends with email addresses and then only to friends with gmail.com email addresses using a special class construct and a rule engine.

References

[1] Holger Knublach *Ontology-Driven Software Development in the Context of Semantic Web: An Example Scenario with Prot´eg´e/OWL.*
[2] W3C Working Group (2006) *A Semantic Web Primer for Object-Oriented Software Developers.*
[3] W3C (2001) *Semantic Web Activity home page.*
[4] Brachman R.J. (1983) *IEEE Computer*, 16(10), 30-36.
[5] Fensel D. (2001) *Ontologies: Dynamic networks of formally represented meaning*.