A NOVEL CONSTRUCTION OF ONTOLOGY DESIGN FOR CODE CLOINING

[1]Syed Mohd Fazalul Haque
Maulana Azad National Urdu University
fazal.manuu@gmail.com

[2] Dr. V Srikanth
Professor, Dept of CSE, K L University
srikanth_vemuru@yahoo.com

[3]Dr. E. Sreenivasa Reddy
Professor, Dept of CSE, ANU
esreddy67@gmail.com

Abstract - Code plagiarism is a serious and ongoing problem in the area of computer science which affects the quality of the code and software developed. This type of problem occurs mainly due to rapid development of software without following certain paradigms. In recent years, numbers of software developers are copying the code from the open source code available on the internet database. Verifying the written code manually is a very difficult and to identify the similarities is a labor task and longtime consumption. It may be likely to be impossible and difficult due to the availability of larger repositions. Ontology is an approach which is used to describe the semantics of the documents, which can be used as a file for source code too.

Web ontology language (OWL) is self-defined application code which describes taxonomy and vocabulary of code related to programming language.We use web based SARQL an SQL query language which extracts information from the save ontology for identification of clones in code.We propose a code cloning detection method based on created ontology using editor Preodwhich is used to identify code developed and cloned.

Keywords: PREOD,Plagiarism, OWL, Ontology,SARQL

1. Introduction

In present days very huge volumes of information is stored in digital form which has many advantages and dis-advantages comparatively.As the information is available when needed more quickly (by clicking a button or execution of an event), this is the advantage of cloud and cloud repositions.

In the present days software code is available open source in digital repositions, this code is reused in developing new software without altering the functionality of the code. These may lead to identify the original software with that of duplicate software. The open source code have advantages and disadvantages.

In relate to the disadvantage, there lies a conflict in identifying the original with duplicate document, as and when the operation is done manually. This is the case we are trying to identify clone detection system for identification of duplicate or copied clone.

The word “ontology” is related to the reference of existence and the one which exist.

In the field of code cloning information, ontology and data are expressed in terms of semantics and syntax related to the domain specified (related to a programming language), it gives a vocabulary to the domain and also generalizes the computer meaning used for vocabulary.

Ontology’s assorts from classification, taxonomies, and schemas of database in the form of theories which is axiomatized.In future ontology’s and present many scientific and business organizations are using it for sharing, reusing the knowledge domain.Ontologies are been the framework now a days in running the business effectively for various application in knowledge sharing, integration , services of web, management of information and commerce services.

We propose and use ontology for gaining knowledge graph in identifying the code which has been cloned or reused in software development. Web ontology language (OWL) gives us a specification of World Wide Web Consortium (W3C), which provides us a basic component of Web semantic initiation.

OWL is a tool which is based on XML. XML used in ontology framework uses resource description framework (RDF) and Schema RDF.

This xml is composed from 3 sub-languages namely OWL-Full, OWL-DL and OWL-lite, of all these, DL-OWL is the one which is commonly used and also provides use the best expressiveness. RDF(Resource Description Framework) language used to provide information about the resources in WWW.It also provides use information about the meta data of resource namely author, title, date of hosting the page and last modified date, license agreement copyright form and the schedule of the resource.

This RDF provides use the information about the related things or concept directly or indirectly using a web resource based on the semantics in OWL.

It also provides us the information needed for code cloning and application processing even thou the user doesn't require the necessary.

It also gives a generic framework for delivering the information and also exchanges between the applications without losing the meaning. As the common generic framework used by the designer can force the use of global RDF parsers and tools used for processing. It also provides use to exchange the data between various applications and are also available to the applications for which they are created originally.

We proposed to use OWL and RDF method and their standard formats for preparing ontology and creating using editor Preod.W3C standards in using this approach, this approach provides use interoperability among the older and the future related works.

Preodis an ontology editor which is developed using web tools; it provides us the framework based on knowledge to build domain and application related to knowledge based on ontologies.It is also a base for implementation of modeling actions and structures with the basic support of visualization, creation and manipulation related to ontology and formats representation.It is customized and user-friendly support for creating domain knowledge on models and data entry into it.

RDF for SARQL is a query processing language, which is used to retrieve information from various sources from a native RDF storage or RDF middleware.SARQL has query capabilities in generation graph optimal pattern based on disjunctions and conjunctions.

It also extensible support constraining and value testing query based on RDF source graph. The results of the query are shown in the form of a graph RDF.

2. Proposed Method

Our proposed work is narrated in algorithms steps with Preod editor of version 4.3.0 it is platform independent and it has a support to ontology modeling. It has a built in editor which enables the user to built generic ontology framework in accordance with open source knowledge connective protocol.The model uses ontology, which consist of various classes and hierarchy between classes which forms a relation and associated between relationships. It also describes the property of instances and number of individuals among the domain class.

OWL editor Preod enables the user to develop own ontology's based on web semantic and also uses W3C integrated with ontology (OWL). The ontology OWL will include classes, description of classes, properties and instances of classes.The formal OWL ontology specifies the semantic of the code which is derived logically and consequentially based on facts available in ontology.

The mapping of the code is done based on single document or multiple document which is distributed and combined based on OWL mechanism.As stated above, the second way used in ontology modelingis provided by Preodwhich uses OWL and W3C base ontologies.The first pace in detection of code cloning system is to build an OWL structure class. This part is similar to object oriented programming paradigm.

Table 1 . Ontology of Clone instance and Clone class

Subject	Relation	Object
Clone set	Contain	Clone instance
Clone instant	Reside_in	Method
Clone instance	Diff_use	Function, field, variable, class, interface,
Clone instance	Common_use	Variable, loop, functions, class, interface
Class	Extend	Class
Class	Implement	Interface
Class	Declared_in	Class
Interface	Extend	Class
Method	Declared_in	Interface
Method	has_return_type	Class/interface
Field	Has_type	Interface
Field	Has_type	Class/interface
Field	Has_type	Method, for, while , class, interface

The class is implemented using Data types, Constant Variables, Structure programming comments, Functions of system and Operators related to.The created structure class is visible using an editor as shown in figure 1 and table 1.

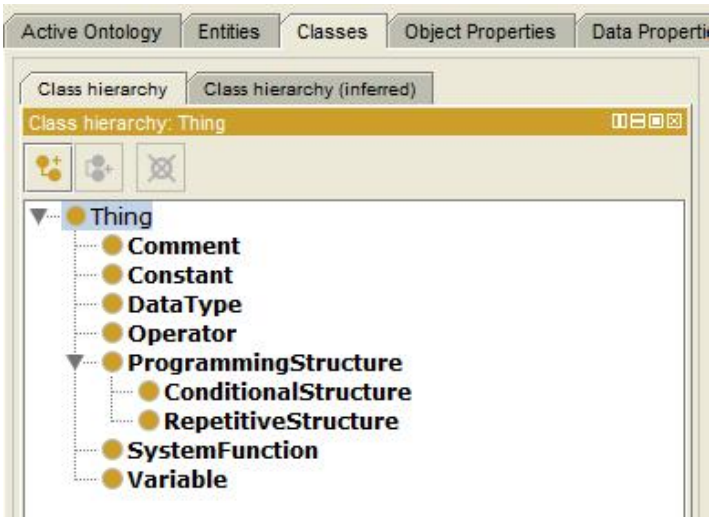

Figure 1 Ontology of Class OWL

Syntax of OWL class is shown above

```
<owl:Classrdf:about="&untitled-ontology-
4;Comment"/><owl:Classrdf:about="&untitled-
ontology-
4;Constant"/><owl:Classrdf:about="&untitled-
ontology-
4;DataType"/><owl:Classrdf:about="&untitled-
ontology-4;Operator"/>
<owl:Classrdf:about="&untitled-ontology-
4;ProgrammingStructure"/>
<owl:Classrdf:about="&untitled-ontology-
4;SystemFunction"/><owl:Classrdf:about="&untitled-
ontology-4;Variable"/>
```

We specify the concept of building the taxonomies of structure which is repetitive and conditional as shown in figure 1.There are also defined as programming structures of special category using structure programming.

The syntax of OWL is shown below as

```
<owl:Classrdf:about="&untitled-ontology-4;ConditionalStructure">
```

```
<rdfs:subClassOfrdf:resource="&untitled-ontology-4;ProgrammingStructure"/>
</owl:Class>
<owl:Classrdf:about="&untitled-ontology-
    4;RepetitiveStructure"><rdfs:subClassOfrdf:resource="&untitled-
    ontology-4;ProgrammingStructure"/>
</owl:Class>
```

For finding the relationship between the model of an object property. These relationships are marked as functional, symmetrical or transitive. Each of the two relationships are marked for inverse. Majority of the relationship are marked as inverse for two relationship for a given property such as propertysubof() in associated with classsubof()

The example below shows the conditional relationship based on Conditional Structure domain based on the repetitive structure range, which is related to another conditional relationship as has_condition.

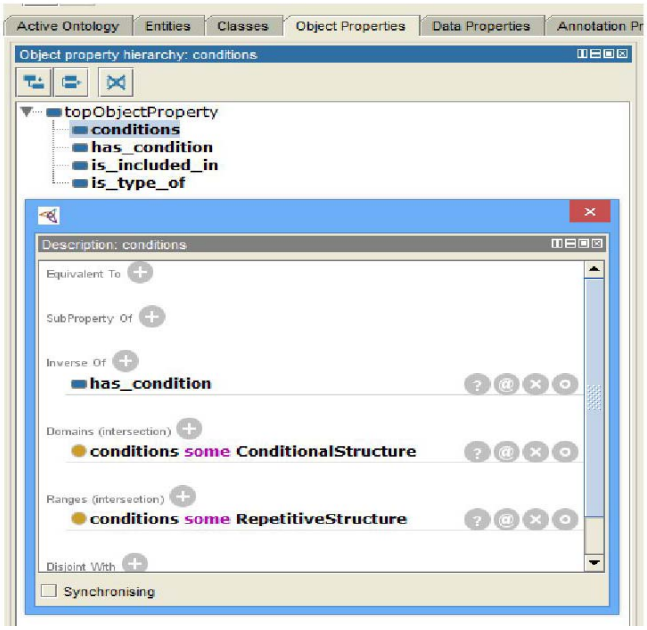

Figure.2. Properties of Object Ontology

OWL syntax for the new concepts is:

```
    <owl:ObjectPropertyrdf:about="&untitled-ontology-
        4;conditions"><rdfs:domain>
    <owl:Restriction>
    <owl:onPropertyrdf:resource="&untitled-ontology-
    4;conditions"/><owl:someValuesFromrdf:resource="&untitled-ontology-
    4;ConditionalStructure"/>
    </owl:Restriction>
        </rdfs:domain>
        <rdfs:range>
       <owl:Restriction>
       <owl:onPropertyrdf:resource="&untitled-ontology-4;conditions"/>
<owl:someValuesFromrdf:resource="&untitled-ontology-
4;RepetitiveStructure"/>
</owl:Restriction>
</rdfs:range>
</owl:ObjectProperty>
    <owl:ObjectPropertyrdf:about="&untitled-ontology-
        4;has_condition"><owl:inverseOfrdf:resource="&untitled-
        ontology-4;conditions"/>
```

```
</owl:ObjectProperty>
```

Other defined relations in our ontology are is_included_in(which is marked astransitive) and is_type_of. We could limit their domain and range as well, toProgrammingStructure or Variable

And DataType. The correspondent OWL syntax for them is:

```
<owl:ObjectPropertyrdf:about="&untitled-ontology-
    4;is_included_in"><rdf:typerdf:resource="&owl;TransitiveProperty"/>
</owl:ObjectProperty>
<owl:ObjectPropertyrdf:about="&untitled-ontology-4;is_type_of"/>
```

The next step is to define facts upon the previously defined concepts, attributes and

relations by instancing them. These instances are called individuals (similar to the OOP concept of object [7]). The following example (Figure 3) states that For and While

Figure. 3. Structure of Individual Code

The specific OWL syntax for individuals is:

```
<owl:NamedIndividualrdf:about="&untitled-ontology-4;While">
    <rdf:typerdf:resource="&untitled-ontology-
4;RepetitiveStructure"/></owl:NamedIndividual>
<owl:NamedIndividualrdf:about="&untitled-ontology-4;For">
    <rdf:typerdf:resource="&untitled-ontology-
4;RepetitiveStructure"/></owl:NamedIndividual>
```

For every source code ontology is created which is suspected for plagiarism.This can also be done manually for the purpose of demonstration using Preod.A crawler reads the source code automatically and used to build a OWL file related to it[8].The main work of crawler is receiving an input as raw source code and gives OWL file as output of which it relates to the predefined ontology. So that every source code is having the ontology file irrespective of the language of which it written .Here is a small source code which was written in C

```
int option = 0;
int i;
int numbers[3];
while (option!=3)
{
      printf("Please choose an option and press enter:\n");
      printf("1. Read 3 numbers\n 2. Print the max\n
      3.Exit\n"); scanf("%i",&option);
      if (option==1)
      {
            for (i=0; i<3; i++)
            {
                printf("\nnumbers[%i]=",i+1);scanf("%i",&numbers[i]);
            }
      }
      if (option==2)
      {
            int max = 0;
```

```
                for (i=0; i<3; i++)
                {
                      if(numbers[i] > max)
                      {
                            max = numbers[i];
                      }
                }
                printf("\nMax=%i",max);
          }
    }
```

Here the code has three different items

1. Input read three numbers
2. After reading the numbers print the maximum number
3. Finally exit
   All these can be displayed in a menu

From the given integer values among three which was given by the user, computes the maximum number or otherwise execution failed for incorrect data has read.This process is automatically done by the crawler which was discussed in the below paragraph. Generally the crawler reads the code from top to bottom line by line and generates specific individuals for every line

For example:

(i)The first three lines describes the type of variable which gives property of the value given.So the object property is_type_of is assigned for individuals of DataType namely array &int .

(ii) Next lines contain the programming structure with different individuals of type SystemFunctions of which it contains the object property is_included_in.

(iii) Remaining code also applied for the same rule. Finally it finishes parsing of all the source code.similarly all the conditional and repetitive programming structures named the different individuals related to the psedocode applied such as if, if2, switch, while, do-while,for,for2 etc,but we should not use object properties for all the structures

The comparison is done based on the code written in javascript which does the same thing. The similarities of the syntax are checked between the two languages.

```
    var option =
    0; var i=0;
    var numbers=new
    Array();
    while(option!=3)
    {
          document.write("Please choose an option and press
          enter:\n"); document.write("1. Read 3 numbers\n 2. Print
          the max\n 3.Exit\n"); option = prompt("Option");
                if (option==1)
                {
                      for (i=0; i<3; i++)
                      {
                            numbers[i] = prompt("numbers[" + (i+1) +
                            "]");
                      }
                }
                if (option==2)
                {
                      var max = 0;
                      for (i=0; i<3; i++)
                      {
```

```
                        if(numbers[i] > max)
                        {
                              max = numbers[i];
                        }
                  }
                  document.write("\nMax=" + max + "\n");
            }
      }
```

Similar to what we have done before wecreate an ontology for the Javascriptsourcecodetoothe above source code created in java script is check for ontology in code cloning is also measure for similarities.The last step proposed gives or compares of two codes written in c and java script are mapped with the ontologies based on the current process.

The solution gain is verified with the OWL Ontology'sbased on SARQL query in building the code and comparing the code with the source.The algorithms written mainly depend on the query for testing the clone and measure of clone in code. Some metric measures are chosen using SARQL and is compared with the cloning degree of code of software.

SARQL is a SQL to access data of RDF which uses triple pattern in storage and access.Triple RDF has the same pattern form of triple in counter part of SQL database like, join-project-select in Query language. The Query SARQL will support both disjunction and conjunctions concept in triple patterns.

Likewise,predictions of Query of SARQL are also varied and also allow us to state the predication based on predicate-diagnostic.EditorPreod provides an interface to run SQL queries based on results and results sets as shown in figure 4.

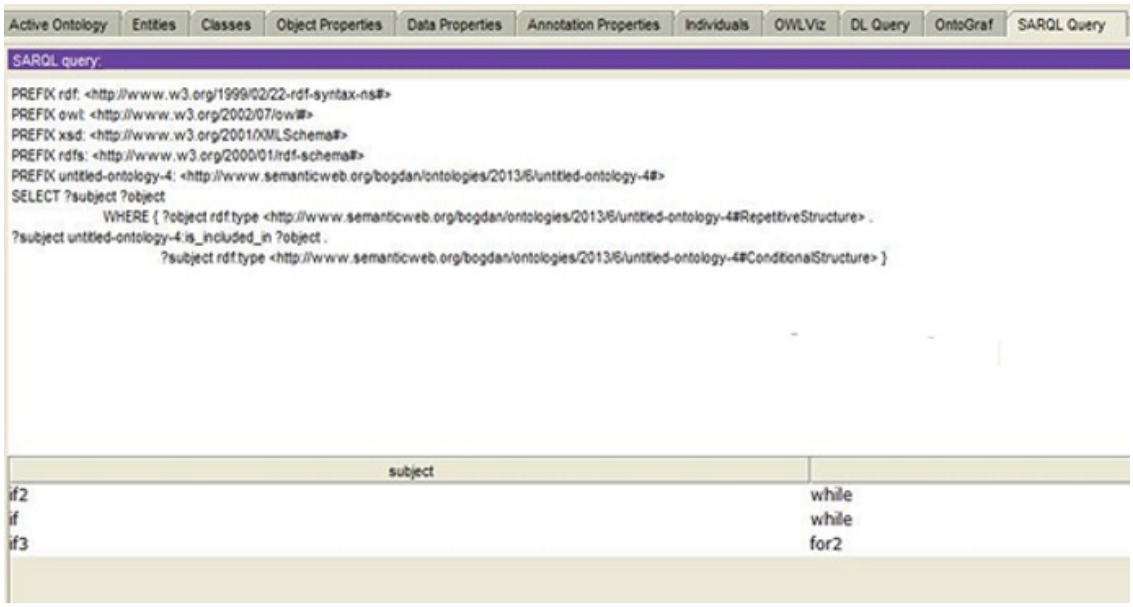

Figure 4. SARQL - SQL EDITOR

We have defined 10 metrics for measuring the ontology as given in Table 1. Each of the metrics are computed based on cloning degree.

Table 1 . Shows the Metric measure of the code Cloned

No	Metric	SARQL query	Result on C code	Result on Javascript code	Percentage of similarity
1	conditional structures count	SELECT ?subject WHERE { ?subjectrdf:type <http://ontology_uri#ConditionalStructure>}	3	3	100%
2	Repetitive structures count	SELECT ?subject WHERE { ?subject rdf:type<http:// ontology_uri#RepetitiveStructure>}	3	3	100%
3	Variables count	SELECT ?subject WHERE { ?subject rdf:type <http://ontology_uri#Variable>}	4	4	100%
4	Conditional structures included in repetitive structures Count	SELECT ?subject ?object WHERE { ?object rdf:type<http://ontology_uri #RepetitiveStructure> . ?subject untitled-ontology-4:is_included_in ?object . ?subject rdf:type<http:// ontology_uri#ConditionalStructure> }	3	3	100%
5	Repetitive structures included in repetitive structures count	SELECT ?subject ?object WHERE { ?object rdf:type<http://ontology_uri #RepetitiveStructure> . ?subject untitled-ontology-4:is_included_in ?object . ?subject rdf:type <http://ontology_uri #RepetitiveStructure> }	0	0	100%
6	System functions called count	SELECT ?subject WHERE { ?subject rdf:type <http://ontology_uri#SystemFunction>}	5	4	80%
7	Count of system functions called conditional structures	SELECT ?subject ?object WHERE { ?object rdf:type <http://ontology_uri#ConditionalStructure> . ?subject untitled-ontology-4:is_included_in ?object . ?subject rdf:type <http://ontology_uri#SystemFunction> }	1	1	100%
8	Count of system functions called in repetitive structures	SELECT ?subject ?object WHERE { ?object rdf:type <http://ontology_uri#RepetitiveStructure> .	4	3	75%

		?subject untitled-ontology-4:is_included_in ?object . ?subject rdf:type <http://ontology_uri#SystemFunction> }			
9	Count data types used	SELECT ?subject WHERE { ?subject rdf:type <http://ontology_uri#DataType>}	2	2	100%
10	Count of variables of type array	SELECT ?subject WHERE { ?subject rdf:type <http://ontology_uri#Variable> .	1	1	100%

The plagiarism degree is used for the method to determine for the metrics which was chosen precisely drawn. The last user is to select the interest metrics which effects the final result .so that we will consider the metrics which made them equal for the final result .For this we can use the software applications to determine the metric result. Moreover this approach doesn't give fair results means that it is not accurate for the acquisition to give the result .Hence, in the field of ontology, we can search for the alternate techniques to maintain better results.

In the semantic approach, there is another method by determining two different source codes ontologies by representing in graphics.This approach sometimes times called as concept network which is a graph. In this network the nodes are called as concepts and the arcs gives the relationship between these concepts

Mainly these semantic networks are based upon the arcs and finally organized into a taxonomic hierarchy. These semantic networks used for the different ideas such as activation spreading, nodes defined as proto objects and also inheritance concept. These are mainly used for large network domains. In some cases these networks are not supportive .the properties which are not expressed simply are general non-taxonomic knowledge, disjunction and negation. Complementary predicates and specialized procedures are used to check and to maintain the relationships throughout our domain. This problem should not consider as a big problem in this method.

Here the topic map represents the different case in the semantic network representation. In this topic map all the information has to be gathered and designed the standards about the topic related to the subject represents in one location. Here there is a link for the other subjects too in this network so that it is called as subject centric.[13]. The graphical representation gives the ontology structure which helps in our method. The latent semantic analysis which is useful in extracting semantic context and this viewed as Bayesian version.Finally, the projection can be applied on two dimensions to create the topic map visualization [14].

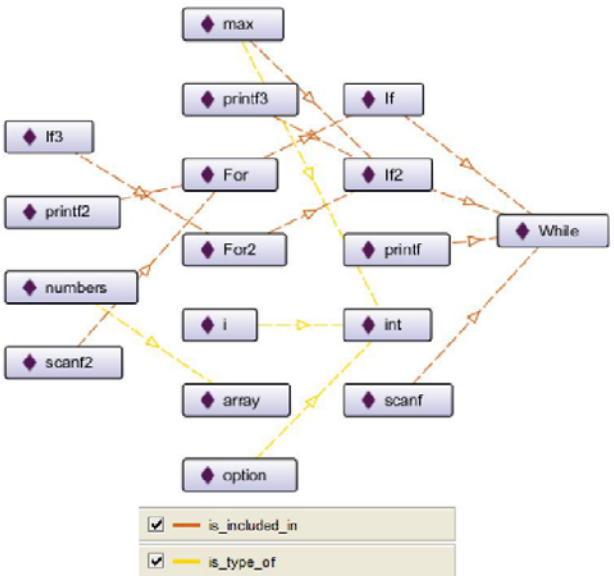

Figure. 5. Ontology of C of individual and representation of OntoGraph

Whenever we want to compare twoontologies, we definitely use Onto graph representation which was discussed in our proposed system.In this figure nodes are represented as violet color diamond in the rectangles .Mainly here two types of relations are there one is represented in yellow color defined as is_type_of relation. And the orange color depicts theis_included_inrelation.The arcs on these links shows on which direction it goes to establish the relation. The hierarchy represents tree horizontal view considered as is_included_in and is_type_ofrelations.

In figure 5 the representation related to c Ontology and the second graph which is shown in figure 6 represents completely java script. Here in this ontology there is different set of individuals, but in C ontology the individuals related to the same set.

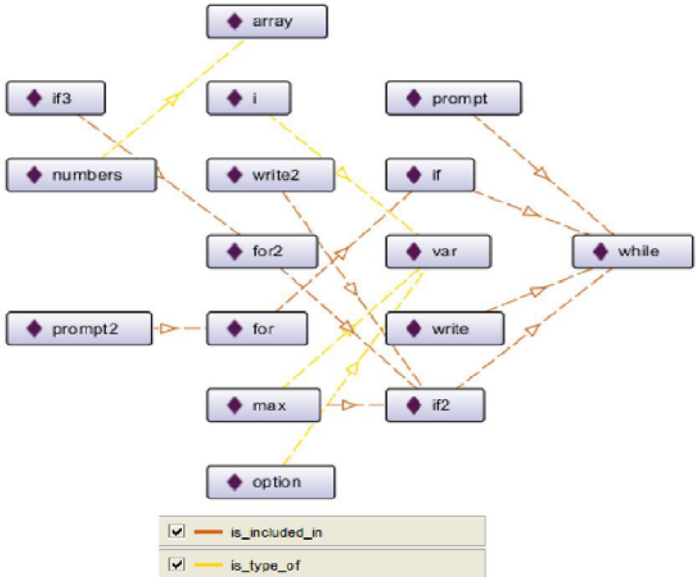

Figure. 6. Ontology of JavaScript–representation of OntoGraph

Whenever we want to compare twoontologies, we definitely use Onto graph representation which was discussed in our proposed system.In this figure nodes are represented as violet colour diamond in the rectangles .Mainly here two types fo relations are there one is represented in yellow colour defined as is_type_of relation. And the orange colour depicts theis_included_inrelation.The arcs on these links shows on which direction it goes to establish the relation. The hierarchy represents tree horizontal view considered as is_included_in and is_type_ofrelations .

In figure 5 the representation related to c Ontology and the second graph which is shown in figure 6 represents completely java script. Here in this ontology there is different set of individuals, but in C ontology the individuals related to the same set.

After comparing the above two topics, we can view them separately and finally we generate a new concept and create a single map represents both the source code ontologies and this new map is completely a plotted graph by looking upon the visualization technique which is an existing tool or by creating a different one.[15]

Even though the representation isdifferent for the similar source codes. So, there is an idea that once the test has done, the source code is copied.Therefore the two forms of plagiarism depends upon the metrics with SARQL and finally topic maps which are more powerful is identifying the similar source codes and on how much percentage. In figure 7 the architecture reveals the necessary steps for us to attain the proposed method.

PROPOSED ALGORITHM

Step 1: First the source code is defined based on the OWL ontology

Step 2: OWL based ontology using SARQL query the RDF graph

Step 3: Based on the result sets metrics are measures

Step 4: Topic map of ontology is represented

Step 5: The final plagiarism degree is determined by finding these results and compares them with the results obtained from ontology.

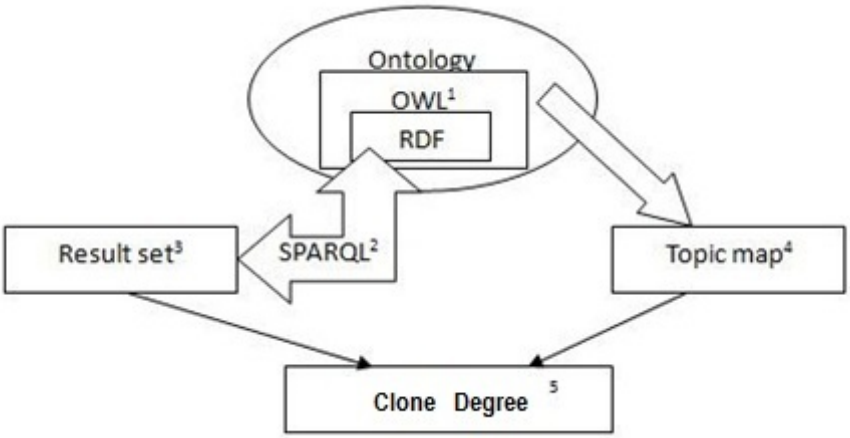

Figure.7. Architecture of the Code Clone detection method

In this architecture, the reliable system of detection had designed all the steps automatically. We have obtained the results based on the metric as shown in figure 8 below .In the future scope, we designed this type of architecture, definitely we will create ontology of OWL, for parsing a crawler and also designed a set of built in/predefined metrics and generated a own SARQL and finally custom representation is done with the above all the topics.These defined components are necessary for the future work.

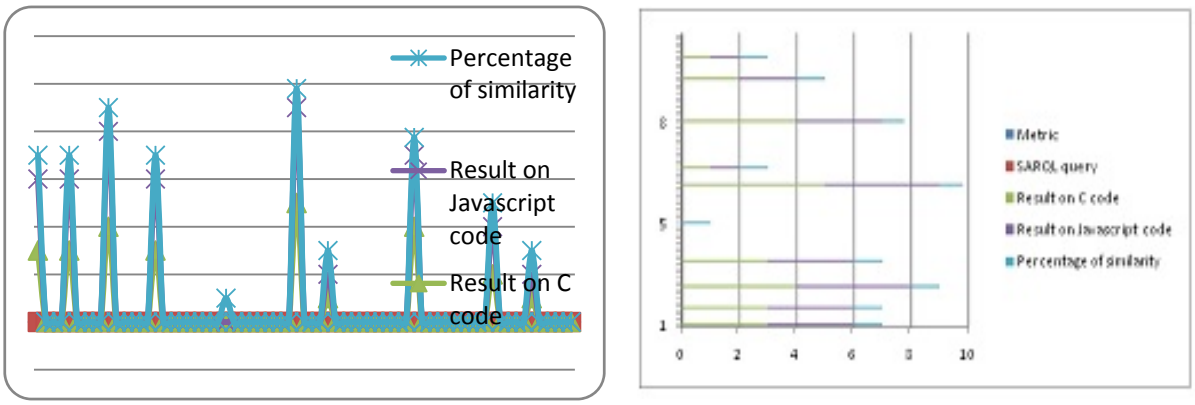

Figure 8.Shows the results of the Metric related to percentage of Code Cloned

Conclusions

our paper gives us a framework for detection of code cloning using ontology, this ise done using OWL a web based ontology language which uses a framework of RDF , this framework is accessed using a developed SARQL query language tool , which extracts related information from the defined ontology based on vocabulary, NLP and taxonomy based on the source code language.We have built an ontology using PREOD a web based ontology editor and SARQL a query processing application which extract and identifies the code clones and its percentage using map. It metric is also measured based on the SARQL query as shown in table above.The main benefit of using ontology is to solve complex software code cloning detection which can be done automatically by query processing. By this we can improve the quality of the code from starting point to end of the code. In future the work can be extended to build automatic tool for code clone detection.

References

[1] M. K. Shenoy, K. C. Shet and U. D. Acharya.(2012, May).Semantic Plagiarism Detection System Using Ontology Mapping.Advanced Computing: An International Journal [Online].3(3). Available: http://airccse.org/journal/acij/papers/0512 acij06.pdf
[2] The Preod Ontology Editor and Knowledge Acquisition System. Internet: http://protege.stanford.edu/ (2013, July 1).
[3] T. Bray, J. Paoli, C. M. Sperberg-McQueen, E. Maler and F. Yergeau. (2004, February 4). Extensible Markup Language (XML) 1.0. W3C Recommendation [Online]. Third Edition. Available: http://www.w3.org/TR/2004/REC-xml-20040204/
[4] F. Manola and E. Miller (2004, February 10). RDF Primer. W3C Recommendation [Online]. Available: http://www.w3.org/TR/2004/REC-rdf-primer-20040210/
[5] S. Harris, A. Seaborne. (2013, March 21). SARQL 1.1 Query Language. W3C Recommendation [Online]. Available: http://www.w3.org/TR/2013/REC-SARQL11-query-20130321/
[6] J. Bao, D. Calvanese, B. C. Grau, et al. (2012, December 11). OWL 2 Web Ontology Language. W3C Recommendation [Online]. Second Edition. Available: http://www.w3.org/TR/owl2-overview/
[7] E Akin, Object Oriented Programming, Houston: Rice University Publishing House, 2001, pp. 33-34.
[8] C. Liu, H. Wang, Y. Yu and L. Xu, “Towards Efficient SARQL Query Processing on RDF Data”, Tsinghua Science & Technology, vol. 15, no. 6, pp. 613–622, December 2010.

[9] I. Ivan and C. Boja, MetodeStatistice in analiza software. Bucharest: ASE Publishing House, 2004, pp. 218-224.
[10] S. Russel and P. Norving, Artificial Intelligence: A Modern Approach (2nd edition). New Jersey: Pearson Education Inc., 2003, pp. 350-352.
[11] D. Newman, T. Baldwin, L. Cavedon and E. Huang, "Visualizing search results and document collections using topic maps", Web Semantics: Science, Services and Agents on the World Wide Web, vol. 8, no. 2-3, pp 169–175, July 2010.
[12] A. Hatzigaidas, A. Papastergiou, G. Tryfon and D. Maritsa, "Topic Map Existing Tools: A Brief Review", in Proc. The International Conference on Theory and Applications of Mathematics and Informatics, Thessaloniki, Greece, 2004, pp 185-201.