Journal of Computer Science 10 (1): 45-53, 2014
ISSN: 1549-3636

doi:10.3844/jcssp.2014.45.53 Published Online 10 (1) 2014 (http://www.thescipub.com/jcs.toc)

SELECTION OF ONTOLOGY FOR WEB SERVICE DESCRIPTION LANGUAGE TO ONTOLOGY WEB LANGUAGE CONVERSION

[1]Mannar Mannan, J., [2]M. Sundarambal and [3]S. Raghul

[1]Department of Information Technology, Anna University, Regional Centre, Coimbatore, India
[2]Department of Electrical and Electronics Engineering, Coimbatore Institute of Technology, Coimbatore, India
[3]Department of Information Technologies, Anna University, Regional Centre, Coimbatore, India

Received 2013-07-19, Revised 2013-10-31; Accepted 2013-11-12

ABSTRACT

Semantic web is to extend the current human readable web to encoding some of the semantic of resources in a machine processing form. As a Semantic web component, Semantic Web Services (SWS) uses a mark-up that makes the data into detailed and sophisticated machine readable way. One such language is Ontology Web Language (OWL). Existing conventional web service annotation can be changed to semantic web service by mapping Web Service Description Language (WSDL) with the semantic annotation of OWL-S. In this conversion of WSDL to OWL process, the ontology plays a vital role. Ontology can be stored and retrieved from local repository and selecting the appropriate ontology is a complicated process and this can be achieved by Ontology Searching and Property Matching (OSPM) engine. Ontology is stored in the local repository as ontology document and exact matching of ontology for the requested query can be searched using semantic similarity ranking method. High ranked classes of ontology will undergo property matching; here requested concept will be matched with the resulting property. OSPM engine act as the backbone for selecting an exact ontology and reduce the conflict that occurs while selecting the ontology for annotation purpose.

Keywords: Semantic Web Service, Ontology Search, WSDL, OWL-S, Property Based Refining

1. INTRODUCTION

As the World Wide Web (WWW) explodes, people have vast amount of information placed at their finger tips. The original WWW consists of document (i.e., web page) and links between documents for additional reference. Cardoso (2007) stated initially WWW planned to develop a universal information database to publish information and that could be accessed in a simple and reliable manner by users. As the days rolls organization started to implement business to customer and e-business over internet. Organization integrate business objective to provide better service to their customers. Finding the relevant services and integrating is a challenging task. After implementing business over internet in a many ways they realized that the initial technology associated with the WWW is not sufficient to integrate many business aspects altogether. Additional functionality is needed to be carrying out the transaction over internet to integrate web services. As per the need HTML evaluated, by so web services are originated.

1.1. Web Services

Web services are modular, self describing, self contained application and software component invoked over the web via an XML message. WSDL defines that the component provides one or more operation to be performed. Currently web services are described by means of a description language, WSDL. This descriptive language has started to be used extensively by many industries such as Amazon, Google and Acrobat and so on. WSDL utilized XML structure and it

Corresponding Author: Mannar Mannan, J., Department of Information Technology, Anna University, Regional Centre, Coimbatore, India

can understand by Human not a machines. In essence, the specification syntax of the input/output message of a basic service, as well as other detail needed for the invocation of the service. WSDL provide the foundation for composition of web service, by providing the support in information exchange between the service, it is not rich enough to specify the semantic of the composition and they are not understand by machine.

Pasupathi *et al*. (2012), focused on segment the content of web document that highly related with query. It is an simple attempt made over the text comparision. Burstein *et al*. (2007) defined well-heeled semantic can support greater automation of service discovery, selection, invocation and automated transformation of message content. To fulfil these needs a new technology need to implement intelligence in automation of service discovery. Many researches are carried out to develop architecture, language that can fulfil the semantic uncertainty in description of content. The result of their work goes under a common heading of semantic web service.

1.2. Semantic Web

Semantic web is a WWW of data base to integrate web data across the world and can enables machine to machine process without human intervention to provide better web services. The major issue in the conventional web service is the problem of discovery and selecting the most suitable web service. To address these challenges the SWS widely used. Farrag *et al*. (2013) provided OSSE, the building block for define the correct ontology.

Web services are described by WSDL and redefined by OWL-S. The existing service can obtained the state of rich semantic automation. OWL-S is itself an ontology which has service profile, service model and service ground as its sub classes. OWL-S is a framework designed for SWS. Farrag *et al*. (2013) shown many web services exist and the service provider needs to change their existing conventional web service to semantic web service. This can be carried out by mapping the important component of WSDL to OWL-S. This process not only converting the conventional web service description language to semantic one, but it standardize of this definition by using the concept of ontology to describe any type of data in the service.

OWL-S differs from the WSDL framework and WSDL lacks in information that the OWL-S needed. To obtain these missing data by manual phase is implemented. The system has an important component called Ontology Search and Property Refining (OSPR) engine which act as the backbone for the autonomous process. In this study, the formation of service profile from the WSDL is concentrated because service profile is employed in discovery process.

This study focused on three areas 1. Describes background knowledge about WSDL 2. OWL-S; 3. describe the mapping model, ontology search, type conversion and refining.

1.3. Web Service Description Language

WSDL combined with Simple Object Access Protocol (SOAP) and XML Schema to provide web services over the Internet. A client process connecting to a web service at the particular instance can read the WSDL to determine what functions are available on the server. A special data types are embedded in the WSDL file in the form of XML Schema. The client can then use SOAP to call one of the functions listed in the WSDL. W3C noted WSDL is a document in XML format for describing the web service. It states the nature and working principles of web service and it is provided by the service provider for service requester. An overview of WSDL document is shown in the **Fig. 1**. WSDL document has abstract and concrete definitions. Data type and element are described in abstract section port and binding operation is explained in the concrete section. Defining the element in abstract section provides a way to reuse the WSDL. The WSDL service is defined using six major elements:

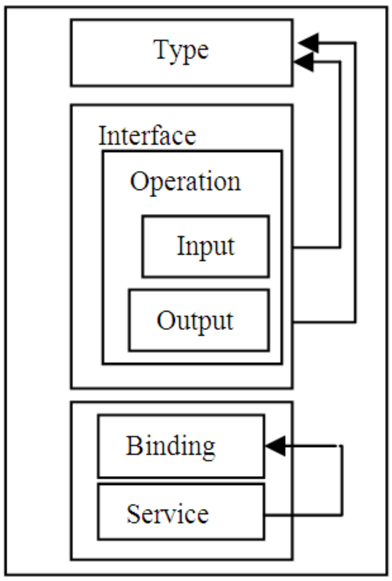

Fig. 1. WSDL component

- Type: Provides data type definition used in messages and Data types are simple and complex type
- Message: This represents the data that is being transmitted
- Port Type: This defines the operation that can be carried out. Each operation refers input and output message
- Binding: It specifies the operation and defines the SOAP and transport protocol
- Port: Specifies address for binding
- Service: Service will act as a container for the functions that have been exposed to Web based protocol

1.4. Web Ontology Language for Services (OWL-S)

OWL-S itself is an ontology used to describe the property and capability of web service in OWL. OWL-S is a W3C submission and many researches are carried out to use it in describing the web service. OWL-S ontology includes the sub ontology's class namely the service profile, used to describe what the service does; the process model is used to describe how the service is used; and the grounding is used to describe how to interact with the service. The service profile and process model are thought to be an abstract characteristic of a service. Whereas grounding makes it possible to interact with a service by providing the necessary concrete detail related to message format. Overall structure of OWL-S is shown in the **Fig. 2**. In this figure the sub concept service profile, process and grounding concept are related to its main service concept with the relations. The relations are presents, support and described by. These relations explain the relation between the service and its sub concept.

The OWL-S service profile has a set of concept to specify the capability and working of the service.

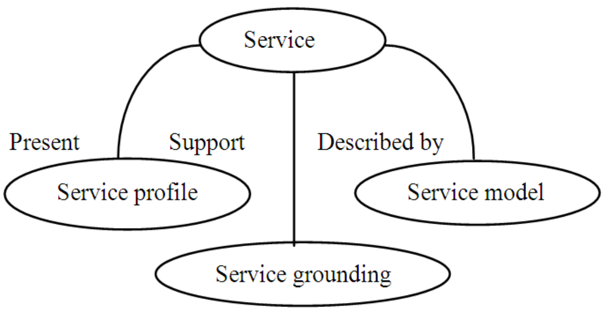

Fig. 2. Structure of OWL-S

The main objective of these concepts is to support high capability of discovery. Service profile information can be categorized into three categories: (1) information that can be readable by human being, it includes the service name, description and the contact information of the service; (2) information that deals with the functionality of the service. This information includes data about input, output, preconditions and effects; (3) information that includes the quality of service.

The OWL-S process specifies the possible pattern of interaction with a web service. Two processes can be invoked atomic and composite process. Atomic process is one in which a single interchange of input and output between a consumer and provider .where as in composite process a set of component process linked together by control flow and data flow. The OWL-S service grounding specifies the details of how an agent can access a service. It includes communication protocol, message format and other service-specific details such as port number used in contacting the service.

2. RELATED WORK

In overall, only some amount of work is carried out in transforming WSDL to OWL-S. Transformation is carried out by annotating the WSDL element with the concept (class) in ontology.

HeB *et al*. (2008), in ASSAM tool are generated to do this annotation process. Web service developers do not sense in terms of ontology's but rather in terms of their programming language. The key feature of ASSAM tool is the ability to suggest which ontological concepts are used to annotate each element in WSDL file. The backbone for this process is machine learning algorithm. Algorithms need study of data to take the decision. Web services with existing semantic annotation are provided as a study data. With this training data set, the proposed machine learning algorithm will provide the semantic annotation for the new web service. This tool is used to generate OWL-S from the WSDL. This research paper provided an origin for this research work to be carried out and insists enhancement needed to autonomous process by machine. Even though this tool provides a way for annotating the WSDL element with the ontology concept, they lack in some area. They are:

- In this tool, process of finding the possible concept for an element in a WSDL file is just a text based search process
- The way of storing the ontology's is not provided and hence complexity is increased when huge numbers of ontology's are referred

- The list of most appropriate class that are introduced by the tool at the end is an unordered list

Kogilavani and Balasubramanie (2012), attempt for level semantic analysis for select the necessary information from documents. All the sentences are annotated with aspects, prepositions and named entities.

Segev and Sheng (2012) stated boostraing ontology by continuous analysis of WSDL documents and employs ontology based on concetp and relation. In order to annotate, concept from the WSDL must be matched to concept from an appropriate ontology out of an ontology store. The way of expressing the element in WSDL get differs from the way of expressing the element in ontology. WSDL describe the binding feature of web service and ontology has concepts and its relations. Therefore it's difficult to match directly between two formats. MWSAF proposed to bridge this gap between the two formats. This is achieved by converting two different forms into a common single form by means of schema graph. When this graph is generated concept matching process is carried out. The accuracy of MSWAF is depends on the threshold value. Schema matching approach is replaced by machine learning approach using navie Bayes classification. Since machine learning process is carried out the process of choosing the appropriate concept is faster because matching with ontology is not carried out.

WSDL to DAMLS provide the basic structure of DAML-S description of the web service and saves a great deal of man power. The translation of a complex WSDL document takes about a week of Man time, where most of the time is spent dealing with the syntactic transformation from WSDL to DAML-S and only few hours to construct the composition of process in the process module and compiling the description of the profile. Using this tool the syntactic translation tales less than a minute. The mapping process is carried out in two steps. They are operation conversion and XSD2DAML conversion. In XSD2DAML conversion translation of XSD type into the corresponding DAMLS concept and in operation conversion WSDL operation into DAML-S atomic process. Even though this saves a great deal of man power this approach suffers in limitation:

- The author is forced to assume a 1:1 ration between the XSD element and DAML concept which are unrelated to the available ontology
- The result of the translation produces a complete specification of the grounding and incomplete specification of the process model and profile

In mapping from WSDL to OWL-S Farrag *et al*. (2013), is carried out the result of this process is the OWL-S file with the semantic annotation. In this author provides a clear way of storing ontology and using it. When the ontology is stored in the repository two main processes are carried out. They are structure extraction and keyword extraction. These data will be used further by the other modules. Quix *et al*. (2011) provide a method to measure similarity between ontologies to overcome the first limitation in standardization process is carried out in which ontology are searched by means of linguistic search and the search result is refined by means of structure refining (concept to concept matching). This process is carried out by Ontology search and standardization Engine shortly OSSE. Data's from keyword extraction are used in search process and structure refining is used in refining process. OSSE act as the backbone for this process. This tool is a semi automated tool because WSDL file doesn't have all the sufficient data for the OWL-S. The result of this tool is the fully annotated OWL-S file. The limitations are:

- Temporary ontology that is created is also stored in the repository for the future use. Efficient and exact way of ontology generation is under research and its need knowledge about the ontology
- In refining process concept to concept matching alone is carried out. This may produce a huge set of refined ontology when the repositories are filled with huge collection of ontology
- Structure of an ontology is not a stable one to be stored it should be modified in repository when changes are carried out which may affect the semantic annotation

3. MAPPING MODEL

The mapping model proposed in this study is based on the continuous analysis of WSDL file and generating the OWL concept from the simple and complex type elements. With the result the ontology are searched and refined based on the semantics and property that the concept holds. Three main processes that are carried out in his mapping model (1) conversion from XML to an OWL representation (identification of concept and property from the type definition) (2) searching the related ontology from the repository (3) search result is refined based on the property. The overall processes that are carried out in this model are shown in **Fig. 3**.

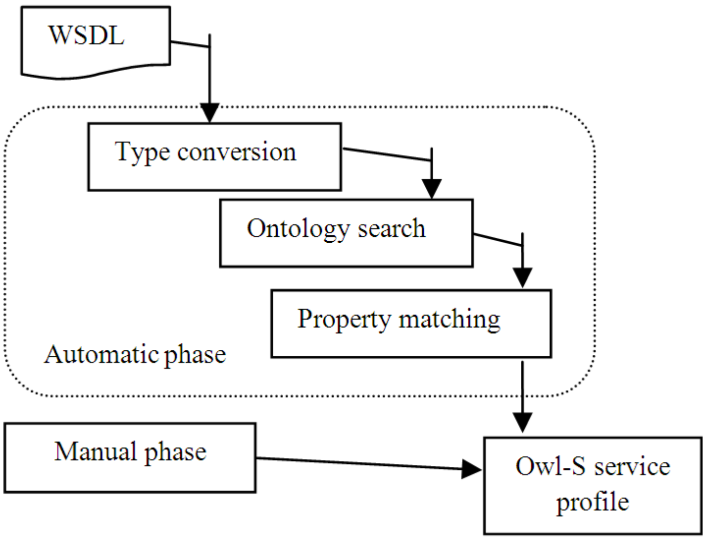

Fig. 3. Mapping model

For the development of the new semantic web service a new ontology can be designed. Designing a new ontology is difficult and time consuming process. Consider a book sales service the ontology refining the concepts in this service may be present in already existing ontology searching an exact ontology for the request concept may reduce the complexity in generating new ontology and ontology reuse can be achieved. Ontology consists of concept, attributes and property representing the relationship between concepts. In many research ontology are searched based on concept and refined based on the structure. Refining the search by the semantic will give high efficient ontology for the search. Each and every block and the process carried out are explained briefly in the upcoming section.

3.1. Type Conversion (XML to OWL)

XML has reached a high position among most standard bodies. There are many reasons for XML to become a highly used standard. The most important reason behind this is its simplicity and suppleness of usage fits well with most part of the application information. The most recent OWL, along with the Resource Description Framework (RDF) on which it is based, has became another poplar standard for data representation and exchange.

The components of the WSDL are described by means of XML tags. Type describes the data that are used in messages. These data may be of simple or complex type and initially the data types are verified whether it is a simple or complex type. Converting a simple data type representation to OWL representation is easy. These converted OWL representation will be used in refining the ontology semantically. The overall process that is carried out is shown in the **Fig. 4**. Nasir and Noor (2011) try to identify commonality of content between revised MT Knowledge Model and IDOC CRM via common conceptualization. The pattern provided by the author is used here for conversion. An example for the complex data type shown in **Fig. 5**.

The complextype name "ListAddressDataset" has three element "countrycode", "postalcode", "postaladdress". This complextype can be redefined by means of OWL this is shown in the **Fig. 6**.

3.2. Ontology Search

After the element is converted to OWL representation we have a concept and property. For that concept exact ontology can be selected from the repository. Farrag *et al*. (2013) mentioned a clear view of storing the ontology in the repository is stated. By using these concepts, the local ontology repository can be developed. Inspect of extracting the ontology structure, OWL tags are removed and concepts with the key words are stored as the document (text file). This document utilized to rank the text based (result) search using TF/IDF Ranking. Local repository has the collection of ontology and the list of concept that it holds. This list is maintained as a document in the repository. Consider the repository has 40 ontologies and 880 the concept it holds in total. The overview of the process is to be carried out is shown in **Fig. 7**.

INPUT: WSDL file
PROCESS:
1. Check single or complex type
2. Convert XML to OWL representation
OUTPUT: concept and property in OWL representation

Fig. 4. Overview of type conversion

```
<s: complexType name="ListAddressDataset">
<s:sequence>
<s:element minOccurs="0" maxOccurs="1"
name="countrycode" type="s:string" />
<s:element minOccurs="0" maxOccurs="1"
name="postalcode" type="s:string" />
<s:element minOccurs="0" maxOccurs="1"
name="postaladdress" type="s:string" />
```

Fig. 5. Complex type declaration

ListAddressDataSet: equivalent
Class[rdf:type PWL:Restriction;]
Owl: onProperty: has_countrycode
Owl: onProperty: has_postalcode
Owl: onProperty: has_postaladdress

Fig. 6. OWL representation

INPUT: Requested concept from WSDL type

PROCESS:
1. Ontology search from repository
2. Using TF/IDF to find the exact matching ontology document.

OUTPUT: Exact concept matched ontology

Fig. 7. Overview of ontology search

In the proposed system, the TF/IDF algorithm is used to match the exact ontology from the list of ontology. TF (Term Frequency) used to calculate the number of occurrences of a term is in a document. This can be represented as tf_t. Where t denotes the term (concept). Cosine formula to find the similarity between the requested concept and the collect of document is shown in Equation (1). Highest similarity document is selected. This reduces the searching result and the complexity to choose the best matching ontology. For tzhe requested concept related keywords and similarity are obtained from the WordNet, stated in Hong (2011). This increase the possibility of searching the related concepts. Concept Car is related to automobile, machine by searching with the related word increase out search result.

3.3. Property Based Matching

Property distinguishes the class from other classes. For an example truck and car comes under the concept vehicle. Even though, these two classes comes under the same concept with differ functionally. The requested concept has a set of property and the ontology that we acquire by searching also has concept and property. A concept is entirely identical when all the requested property matched with the selected property from the search result.

Requested Concept (RC) has a set of property RP_i, where i = 1,2,3,....n. Concept with the property can represented as RC = $\{RP_1, RP_2, RP_3, ..., RP_n\}$. The set of matching concept is represented as 'C' and also has a set of property is represented as 'CP_j' where j = 1,2,3,...,m. 'm' is the total number of property that 'C' holds.

If two sets have the high degree of similarity then the intersection ($\cap$) between two sets will provide a high value.

To acquire similar property between the requested concept and the concept from the ontology can be measured by:

$$\text{Property Matching} = \left\{\frac{RC \cap C}{n}\right\}$$

where, 'n' is a total number of Requested Property and RC ∩ C, Common property between 'RC' and 'C'. The value for the property match lies between 0 and 1. What steps to be carried out based on the result is shown below:

$$\text{Property Matching} = \begin{cases} 0, \text{no common property} \\ >0.5, \text{property can be mapped} \\ 1, \text{all property presented} \end{cases}$$

If the property match zero ensures no common property between requested concepts and ontology. The corresponding concept dropped and next concept should be checked for relevance. If the result obtained is 1 shows the exact matching of property. The result greater than 0.5 shows the portions of properties are available and some properties are not available. The missing property can be added to the concept.

4. EXPERIMENTAL CALCULATION

4.1. Input File: Weather Fetcher WSDL

Repository Content of all OWL file is converted to a document file (text file) by removing OWL tags and comments and each file considered as separate document. These documents are stored during the repository import process. The appropriate ontology is identified for convert WSDL file into WOL ontology. The ontological text document is used for compare with WSDL for similarity measurements. Various similarity calculations between document and WSDL files carried out by the following techniques.

4.2. Type Conversion

WSDL file is parsed then complex and simple type elements are extracted. Type element is considered as concept name and sequence elements are considered as property:

Concept: Weather.
Property: Time, temperature, wind freeze, humidity. Searching using TF/IDF and calculating TF are given in **Table 1**.

4.3. Normalization TF

To normalize the 'tf' weights of all terms occurring in a document by the maximum 'tf' in that document. In a document the highest term occurrence is divided by all other terms in the entire document.

D_i denotes Document, where i = 1, 2, 3 ...n and t_j denotes term, where j = 1, 2, 3.....n:

$$\text{Normalized tf} = \left(\frac{tf(t, D)}{\text{Highest } tf(j) \text{ in } Di}\right) \quad (1)$$

For every document, each value from the TF in **Table 1** is divided by the maximum value from that document. Resulting value is shown in the **Table 2**.

4.4. Inverse Document Frequency (IDF)

IDF provides weight to distinct terms in a document collection where as document frequency provides weight to frequency of terms in a particular document. The IDF can be calculated by:

$$IDF_t = \log\left(\frac{Dn}{DF_t}\right) \quad (2)$$

where, 'Dn' is a total no of document in the collection and 'DF_t' represent no of document containing t. By substituting the value in Equation (2) and compute IDF for all the documents are $IDF_1 = IDF_2 = IDF_3 = IDF_4 = 0.125$.

4.5. Calculating TF*IDF

To identify the composite weight of the each term in a document, TF and IDF is combining together. **Table 3** shows the composite weight of each concept present in each document.

4.6. Vector Assigning

'Q' is the vector of the term and QIDF is the product of vector and IDF concept term vector is set 1 and the remaining term set as 0. The documents are ranked based on the concept so concept term alone is assigned 1.

4.7. Similarity Calculation

Using cosine formulae (3) the similarity can be measured and resulted value lies between 0 and 1. 1 represents high degree of similarity and 0 represent low Equation (3):

$$\text{Cosine formulae} = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n} A^2} \times \sqrt{\sum_{i=0}^{n} B^2}} \quad (3)$$

Table 1. Term frequency

	D1	D2	D3	D4
Weather	11	58	2	0
Time	2	1	62	5
Temperature	8	1	5	0
Wind chill	9	2	1	0
Humidity	1	1	0	0

Table 2. Normalized TF

	Normalized TF			
	D1	D2	D3	D4
Weather	1.000	1.000	0.033	0
Time	0.182	0.017	1.000	1
Temperature	0.727	0.017	0.081	0
Wind chill	0.818	0.035	0.016	0
Humidity	0.091	0.017	0.000	0

Table 3. Calculating TF*IDF

	TF*IDF			
	D1	D2	D3	D4
Weather	0.125	0.125	0.004	0.000
Time	0.023	0.002	0.125	0.125
Temperature	0.091	0.002	0.010	0.000
Wind chill	0.102	0.004	0.020	0.000
Humidity	0.027	0.005	0.000	0.000

Table 4. Document and Ontology

Document	Ontology
D1	WeatherConcept.owl
D2	FastWeatherConcept.owl
D3	Sumo.owl
D4	Time_entry.owl

Table 5. Vector IDF

	Q	QIDF
Weather	1	0.125
Time	0	0.000
Temperature	0	0.000
Wind chill	0	0.000
Humidity	0	0.000

Table 6. Similarity comparing

Document 1		Document 2		Document 3	
A	B	A	B	A	B
0.125	0.125	0.125	0.125	0.004	0.125
0.023	0.000	0.002	0.000	0.125	0.000
0.091	0.000	0.002	0.000	0.010	0.000
0.012	0.000	0.004	0.000	0.002	0.000
0.027	0.000	0.005	0.000	0.000	0.000
0.667		0.998		0.030	

Table 4-6 shows the cosine similarity measurement of all the document and in D4, the concept term value is 'zero' and it is not necessary to calculate similarity match. Document D2 has the value nearby to 1. D2 has the concept related to our requested concept weather.

4.8. Property Matching

Now d2 has the concept related to the requested concept Weather.

Let us consider RC, Requested concept RP_{i} and where i = {1,2,3,......,n}, Requested property.

Concept can be defines as a set holding the elements i.e., properties Equation (4):

$$RC = \{RP_1, RP_2, RP_3, RP_4\}$$

$$\text{Property Matching } = \left\{\frac{RC \cap C}{n}\right\} \tag{4}$$

where' n' is a total no of Requested Property $RC \cap C$ common property between RC and C.

Resulting value will lies between 0 and 1:

$$\begin{array}{c}\text{Property}\\\text{Matching}\end{array} = \begin{cases} 0, & \text{no common proprty} \\ >0.5, & \text{missing property can be mapped} \\ 1, & \text{all property presented} \end{cases}$$

Requested concept: Weather = {time, temperature, windchill, humidity).

Concept found: Weather = {error, lastupdate, temperature, time, heatindex, windchill, wind, humidity}.

Get Weather Historical by Zip Type: Declaration = {postcode, date, time, licensekey}.

For 'weather', the Property $\text{match}_{\text{weather}} = \frac{4}{4} = 1$

Property $\text{match}_{\text{getweatherhistorybyziptypedeclaration}} \Rightarrow 0.25$

The concept weather has the high degree of conceptual match. For the requested complex type element weather from the repository fastweather.owl is matched based on TF/IDF similarity calculation and the concept weather is matched based on property. Concept weather can be used to annotate the complex type element in mapping process.

5. EXPERIMENTAL RESULT

In the proposed system, the mathematical calculation is prove that the selection of correct ontology for converting WSDL to OWL.

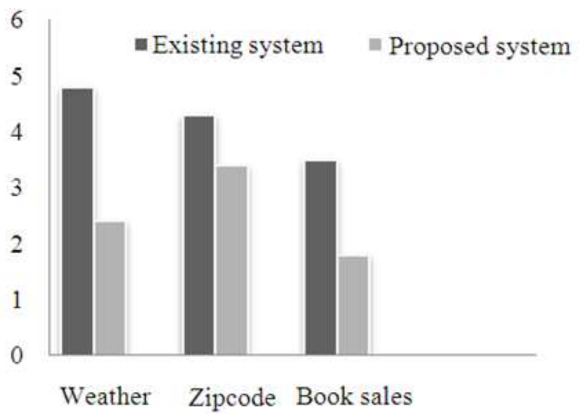

Fig. 8. Resulting graph

Ritcharoenwattu and Rungworawut (2013) Provides a frame for extract new pattern using mining association rules. For a WSDL file, type elements are extracted and respective concepts are recognized. The outcome Result is compared with the existing system is shown in **Fig. 8**. The concepts that are searching and identified are the weather, zip code and book sales. The repository has a corpus of ontology document. The **Fig. 8** shows the performance analysis between the existing and proposed system in obtaining the numbers of ontology at end of the request. The proposed model ensures the reduced number of concept obtains in our proposed system and this will reduces complexity of selecting ontology for requested type elements. Here the not only concepts are searched alone; but searching carried out by considering the property also. By doing so, the search result that we acquired is reduces and also concentrate on the semantic nature of the requested concept.

6. CONCLUSION

In this study, we targeted to solve the problem of complexity in choosing the exact ontology for the type elements in the mapping process from WSDL to OWL-S. It has been achieved by using the TF/IDF and the property matching.

In the proposed system, selection depends on the OSPM component that uses the local ontology repository. By removing the OWL tags and storing the ontology concept with its keyword as a document helps in calculating the semantic similarity for requested concept and also extraction of concept and its property facilitates the property matching. Unlike the previous work this proposed system doesn't depend on the structure of ontology.

Experimental result shows that the proposed system is capable of matching and reduces the complexity in finding out the exact ontology from corpus of ontology.

7. REFERENCES

Burstein, M.M., M. Burstein, D. McDermott, S. Mcilraith and M. Paolucci *et al*., 2007. Bringing semantics to Web services with OWL-S. World Wide Web, 10: 243-277. DOI: 10.1007/s11280-007-0033-x

Cardoso, J., 2007. Semantic Web Services: Theory, Tools and Applications. 1st Edn., Information Science Reference, Hershey, ISBN-10: 159904045X, pp: 353.

Farrag, T.A., A.I. Saleh and H.A. Ali, 2013. Toward SWSs discovery: Mapping from WSDL to OWL-S based on ontology search and standardization engine. IEEE Trans. Knowl. Data Eng., 25: 1135-1147. DOI: 10.1109/TKDE.2012.25

HeB, A., E. Johnston and N. Kushmerick, 2008. ASSAM: A tool for semi-automatically annotating semantic web services. Proceedings of the 12th International Conference on Web Technologies, Nov. 7-11, Springer Berlin Heidelberg, Hiroshima, Japan, pp: 470-475. DOI: 10.1007/978-3-540-30475-3_23

Hong, J.L., 2011. Data extraction for deep web using wordnet. IEEE Trans. Syst. Man Cybernet., 41: 854-868. DOI: 10.1109/TSMCC.2010.2089678

Kogilavani, A. and P. Balasubramanie, 2012. Sentence annotation based enhanced semantic summary generation from multiple documents. Am. J. Applied Sci., 9: 1063-1070. DOI: 10.3844/ajassp.2012.1063.1070

Nasir, S.A.M. and N.L.M. Noor, 2011. Automating the mapping process of traditional malay textile knowledge model with the core ontology. Am. J. Econ. Bus. Admin., 3: 191-196. DOI: 10.3844/ajebasp.2011.191.196

Pasupathi, C., B. Ramachandran and S. Karunakaran, 2012. Web document segmentation using frequent term sets for summarization. J. Comput. Sci., 8: 2053-2061. DOI: 10.3844/jcssp.2012.2053.2061

Quix, C., P. Roy and D. Kensche, 2011. Automatic selection of background knowledge for ontology matching. Proceedings of the International Workshop on Semantic Web Information Management, Jun. 12-16, ACM Press, New York, USA. DOI: 10.1145/1999299.1999304

Ritcharoenwattu, T. and W. Rungworawut, 2013. New patterns discovery for web services composition from mining execution logs. Int. J. Comput. Electr. Eng., 5: 88-92. DOI: DOI: 10.7763/IJCEE.2013.V5.670

Segev, A. and Q.Z. Sheng, 2012. Bootstrapping ontologies for web services. IEEE Trans. Services Comput., 5: 33-44. DOI: 10.1109/TSC.2010.51