Web Service Similarity with Standardized Descriptions

Budi Harjo
Department of Informatics Engineering
Faculty of Computer Science
Universitas Dian Nuswantoro
Semarang, Indonesia, 50131
budiharjo@dosen.dinus.ac.id

Riyanarto Sarno
Department of Informatics Engineering
Faculty of Information Technology
Institut Teknologi Sepuluh Nopember
Surabaya, Indonesia, 60111
riyanarto@if.its.ac.id

Siti Rochimah
Department of Informatics Engineering
Faculty of Information Technology
Institut Teknologi Sepuluh Nopember, Surabaya Indonesia
60111
siti@its-sby.edu

***Abstract*— Nowadays, a web service is one of technologies that is reliable and promising in business practices. This is evidenced by the services created by the providers to support them.Web service similarity is very important to get a replacement Web service, when the web service used is disable. Web service similarity based on annotations and descriptions in this decade are being carried out. Unfortunately, annotations and descriptions given are not standard, so that the Web service similarity does not get the right web service. Furthermore, we propose standardized descriptions obtained directly from the source program of Web service that is generalized into a form WSDL file. This is very important to get a Web service that we are really looking for.**

Keywords - web service similarity; descriptions; operator.

I. INTRODUCTION

In order to develop an information system of a company, the companies have used the information technology called Web service. The researchers have also participated in its development [1] Exchange of Data[2], Web services Discovery, Web service security [3], Accomplishment Evaluation and Composition [4].

The Web Service based on Oriented Architecture (SOA) [5] allows users to discover and compose Web services available. Keep in mind that the web service is provided by the providers in the form of provision represented in a WSDL file (Web Service Description Language). This triggers other methods, for example, OWL-S, Web Service Modelling Language (WSML) [6,7], Web Service Modelling Ontology (WSMO) [7], Semantic Annotations for Web Services Description Language (SAWSDL) [8], and Web Services Description Language Semantic (WSDL-S) [9].

Among these methods, method SAWSDL is the method most widely developed by the researchers, and has also been recommended by the W3C. There are two kinds of Annotations of SAWSDL that are ModelReference and SchemaMapping [10]. They are used generally to anotate WSDL components, and WSDL Type Definitions, and to specify mappings between WSDL Type Definitions in XML and semantic data, and between WSDL Type Definitions in XML and semantic data.

However, in our research, we have found a problem in Semantic Web service discovery using the SAWSDL method with non-standard annoatations, where annotations are given based on the desire of annotations giver. This would create a serious problem when two web services that both have been annoted or described in two WSDL files measured their similarity, for example there are two same web services but have different annotations or two different web services but have same annotations , if they are measured their annotations or descriptions similarity, then we often get inaccurate results or wrong Web service, because their annotations or descriptions given often does not represent their methods. Therefore, we propose a method using Standardized descriptions that can measure Semantic Web service similarity more accurate by marking part of source program of Web Services, then they are copied to WSDL file according their methods position.

II. TECHNOLOGICAL BACKGROUND

A. Web Service Discovery Using many Annotations

Web Service Discovery is required while the users want to find a replacement for a similar Web Service that has been used but it currently can not be accessed to support a business process. The approaches have been proposed by previous researchers to find a Web service using many annotations are. SAWSDL-iMatcher [11], ER Model [12], and A Model-driven [13], and Rule Based Semantic[14].

B. Ontology Web Language-Service (OWL-S)

OWL-S is an ontology that can make functionalities of services become possible. It describe structure of Web Service using three main elements: the service profile to advertise and discover some services; the process model that gives a specialized description of a service's operation; and the grounding that provides specialities on how to operate with one service or several services, through messages [15].

```
package hallo;
public class CalculateNetProfit {
        public String getCompanyName(String CompanyName)
        {
                String CompanyNameResponse;
                //Descripted
                CompanyNameResponse=CompanyName;
                //Descripted
                return(CompanyNameResponse);
        }

        public  Double getCalculateNetProfit(Double grossProfit, Double businessCosts, Double tax)
        {
                Double CalculateNetProfitResponse;
                //Descripted
                CalculateNetProfitResponse=(grossProfit-businessCosts)- ((grossProfit-businessCosts)*tax);
                //Descripted
                return (CalculateNetProfitResponse);
        }
}
```

Fig. 1. Source code of web service is given the remarks

```
<wsdl:operation name="getCalculateNetProfit">
    <xs:documentation xml:lang="en">
      CalculateNetProfitResponse=(grossProfit - businessCosts) - (tax *(grossProfit - businessCosts) );
    </xs:documentation>
 <wsdl:input xmlns:wsaw="http://www.w3.org/2006/05/addressing/wsdl" message="axis2:getCalculateNetProfitMessage"
 wsaw:Action="urn:getCalculateNetProfit" />
<wsdl:output message="axis2:getCalculateNetProfitResponse" />
</wsdl:operation>
<wsdl:operation name="getCompanyName">
    <xs:documentation xml:lang="en">
      CompanyNameResponse=companyName;
    </xs:documentation>
 <wsdl:input xmlns:wsaw="http://www.w3.org/2006/05/addressing/wsdl" message="axis2:getCompanyNameMessage"
```

Fig. 2. Described WSDL

C. Weighted Directed Acylic Graph (wDAG)

The definition of wDAG is the following a normalized arc-labeled, arc-weighted DAG is constructed from a 6-tuple (V, E, LV , LE, LW, r) of a set of nodes V, a set of arcs E, a set of arc labels LV , and a set of arc labels LE, a set of arc weights LW □ [0,1], and one element r □ V, respectively, such that (V, E, LV , LE, r) satisfies the definition of an arc-labeled DAG and there is a fanout-weight normalized many-to-one mapping from the elements in E to theelements in LW, i.e. the weights of every fan-out add up to 1.0. The wDAG calculates the similarity wDAGsim (g, g') from bottom to top recursively. The similarity calculation will be done if two services have same name, otherwise the same calculation wDAGsim (g, g ') will be done. It is clearly formulated as follows [16] :

$$
= \begin{cases} 0, \ \text{if the node label g and g}' \ \textit{are not identical} \\ 1, \ \text{if g and g}' \text{ are leaf node} \\ \sum \begin{cases} wDAGsim(g_i, g_j^{'}).\frac{(w_i+w_j^{'})}{2}, \ g_i \ and \ g_j^{'} \ are \ not \ missing \\ wDAGsim(g_i, \varepsilon).\frac{(w_i+0)}{2}, \ g_i \ is \ missing \ in \ g_j^{'} \\ wDAGsim(\varepsilon, g_j^{'}).\frac{(0+w_j^{'})}{2}, \ only \ g_j^{'} \ is \ node \end{cases} \\ \sum_{j=0}^{breadth\_of\_g^{'}} wDAGsim(\varepsilon, g_j^{'}).\frac{(0+w_j^{'})}{2}, only \ g_i \ is \ leaf \ node \\ \sum_{i=0}^{breadth\_of\_g} wDAGsim(g_i, \varepsilon).\frac{(w_i+0)}{2}, only \ g_j^{'} \ is \ leaf \ node \end{cases} \tag{1}
$$

where, wDAGsim(g, g’): similarity of two input wDAGs g and g’. wDAGsim(g_i, g'_j): intermediate similarity of the ith and jth sub-wDAGs of the wDAGs g and g’, respectively. w_i

and w'$_j$: arc weights of the ith and jth child of the wDAG g and g', respectively.

i: increase from 1 to the breadth of g.
j: increase from 1 to the breadth of g'.
ε: an empty wDAG.
We use this method to depict the structure of our Web service as Fig. 4 and Fig. 5. We use the wDAG method because there is a leaf node that can be reused by other branch.The Tree method cannot used like it.

D. Cosine Similarity

In Cosine similarity method there are two document types [16]. First type is Occurence document and second type is Query document [17]. Occurence document is described as vector form as seen below :

$$\vec{d}=(w_{d0}, w_{d1}, ..., w_{dk}) \quad (2)$$

and query is described as a vector form as:

$$\vec{q}=(w_{q0}, w_{q1}, ..., w_{qk}) \quad (3)$$

where w_{di} and w_{qi} (0≤1≤k) are float numbers indicating the frequency of each term inside a document, while each vector's dimension corresponds to a term available in the document.

Based on vector similarity, similarity between two vectors can be defined as below :

$$Sim\ (\vec{q}, \vec{d}) = \frac{\vec{q}.\vec{d}}{|\vec{q}|.|\vec{d}|} = \frac{\sum_{k=1}^{t} w_{qk}\ x\ w_{dk}}{\sqrt{\sum_{k=1}^{t}(w_{qk})^2}\ x\ \sqrt{\sum_{k=1}^{t}(w_{dk})^2}} \quad (4)$$

E. Wu Palmer Similarity

The Wu Palmer similarity method was proposed by Martha Palmer and Zhibiao [18]. They calculated a similarity of different concepts located in the grammer who performed a measurement avoid distance of concepts to same root concept. In this research, we obtain numbers of similarity between two words using the site http://ws4jdemo.appspot.com/.

III. STANDARDIZED DESCRIPTIONS

Until recently, the WSDL descriptions are given with no standard format, in other words there is no standardization in giving the WSDL description. It causes a different understanding of similar Web service. Finally the Web service similarity finds the web service that does not comply with the demand. Therefore, we propose a concept of giving the descriptions obtained from the source program of Web service depicted in Fig. 1 and Fig. 2, Its Steps for Web service similarity with Standardized we depict in Fig. 3. We started making Class in advance in order to create a Web service that we described in Fig. 1, there we give remarks for flanking operations using the text "// descripted". After we publish Class become a Web service, then a WSDL file without a description of the operation is created, so that the description can be given in standards and in accordance with the operation, then we will create a tool that can copy it to the description of the existing operations in related WSDL file, we expect the result as Fig. 2. Then, we save the WSDL file. In order to obtain the elements for example, service name, input type, and others such as in Table 1,we parse WSDL file, then we save the elements into the database server as metadata of WSDL file. We will use them to calculate the similarity between WSDL file to get WSDL needed.

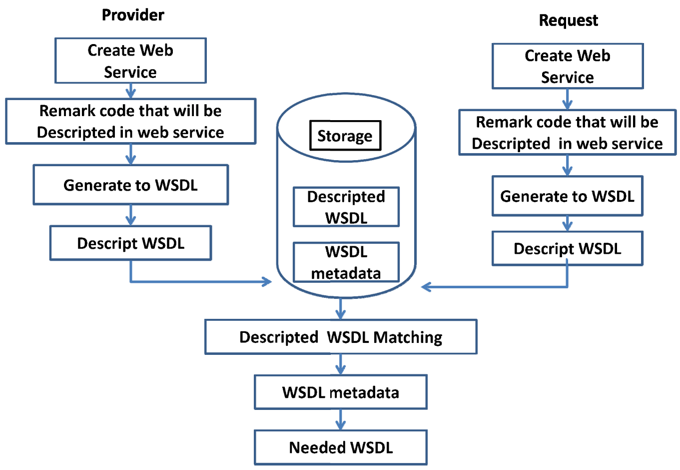

Fig. 3. Standardized descriptions

IV. RESULTS OF EXPERIMENT

We use the conditioned data for the Web service used to calculate the net profit of the company. In this case, we create a main Web service that we give a code "A111", and seven variants Web services of Web service "A111" i.e. "A110", "A101", "A100", "A011", "A010", "A001", and "A000". Code "A" is initial code of Web service, the first number is code of service grounding, the second number is code of service model, and the third code is code of service profile. Number "1" shows that there is a same Web service with Web service "A111", otherwise number "0" shows that there is a different Web service with Web service "A111". To find one of the seven variants Web services that can replace Web service "A111", we measure Web service similarity with using the wDAG method, the Wu Palmer similarity, and the Cosine Measure. We measure similarity of structures of Web services with the wDAG method, while the Cosine measure and the Wu Palmer similarity are tools for measuring words similarity of leaf nodes of Web services. Three of eight the Web services we show in table 1, and we depict their wDAG structures in Fig. 4 and Fig. 5. The roots of them are services containing a name of Web service, for example, Calculate Net Profit and Calculate Net Income. As we have explained in section B, a web service includes of Service grounding, Service Model, and Service Profile. While the content of each of them can be seen from Table 1. Fill of the Service models can be composite or atomic, atomic if it only consists one operation, while the composite if it consists of several interrelated operations. In this research use an atomic type. Fill of Description text is obtained from operators in Operation Description.

In order to find string similarity normalization process must be done before, for example stemming and tokenizing. The formula 3 is used to find string similarity the words of leaf node, its example below :

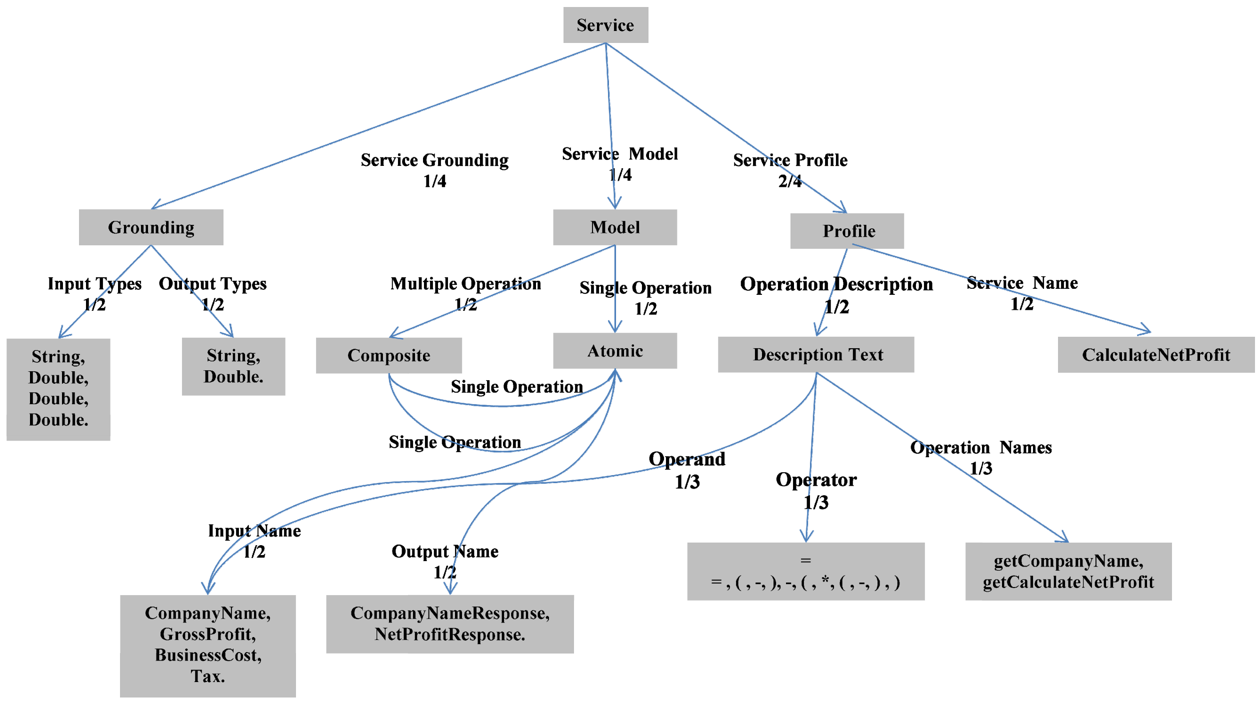

Fig. 4. wDAG Structure of Web service "A111"

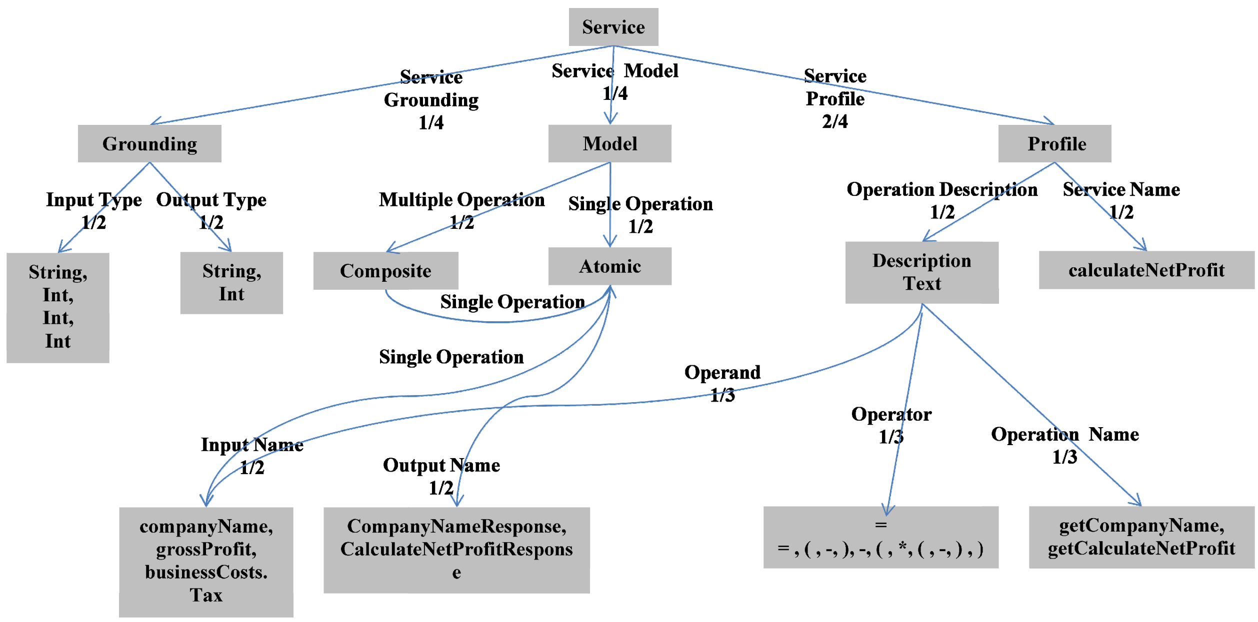

Fig. 5. wDAG Structure of Web service "A011"

TABLE 1. Web Service with Standarized Descriptions

Service Code	Grounding		Model			Profile				
			Atomic			Operation				Service Name
	Input Type	Output Type	Input Name	Output Name 1	Output Name 2	Operation Name 1	Operation Name 2	Operation Description1	Operation Description 2	
A111	String	String	companyName	Company Name Response	Calculate NetProfit Response	Get Company Name	getCalculate NetProfit	CompanyNameResponse =companyName	CalculateNetProfitResponse=(grossProfit - businessCosts) - (tax *(grossProfit - businessCosts))	calculateNetProfit
	Double	Double	grossProfit							
	Double		businessCosts							
	Double		tax							
A110	String	String	companyName	Company NameResponse	Calculate Company Revenue Response	getCompanyName	getCalculate CompanyRevenueResponse	CompanyIdResponse=companyId	CalculateCompanyRevenueResponse=(grossRevenue - businessCosts) - (tax *(grossRevenue- businessCosts))	ComputeCompanyRevenue
	Double	Double	grossRevenue							
	Double		businessCosts							
	Double		tax							
A011	String	String	companyName	Company Name Response	Net Profit Response	Get Company Name	getCalculate NetProfit	CompanyNameResponse =companyName	CalculateNetProfitResponse=(grossProfit - businessCosts) - (tax *(grossProfit - businessCosts))	calculateNetProfit
	Int		grossProfit							
	Int	Int	businessCosts							
	Int		tax							

Word 1 = Double Double Double String
Word 2 = Int Int Int String

$$\text{sim (Word 1, Word 2)} = \frac{\sum_{k=1}^{t} w_{qk} \; x \; w_{dk}}{\sqrt{\sum_{k=1}^{t} (w_{qk})^2} \; x \; \sqrt{\sum_{k=1}^{t} (w_{dk})^2}}$$

$$= \frac{(0.25*0.9)+(0.25*0.9)+(0.25*0,9)+(0.25*1)}{\sqrt{(0.25)^2+(0.25)^2+(0.25)^2+(0.25)^2+(0.25)^2} \; x \; \sqrt{(0.9)^2+(0.9)^2+(0.9)^2+(0.9)^2}}$$

$= 0.999$

Therefore, we put some numbers according to the number of Input Name to find the results of its operations. If the result is same, then we give a value 1, otherwise we give a value 0. After calculating similarity between words of leaf nodes and operators, we calculate their structure similarity using the wDAG method, for example to get similarity of service Grounding between Web service "A111" and Web service "A011" as below :

wDAGsim[Grounding](A111,A011)

$$= \frac{\text{sim [Input Type](A111,A011)+sim [Output Type] (A111,A011)}}{2}$$

$$= \frac{0.999+ 0.999}{2} = 0.999$$

After we count all, then we get the similarity results between Web service "A111" with its variants, as described in Table 2.

Number 0.25 is obtained from number 1 is devided by number of word 1.

Similarity between word Double and Int in Wu Palmer is 0, however in this calculation we give 0.99 because values of them are nearly same.

To calculate similarity among operators, we do not use the Cosine Measure, because if we use it, then there are same operators will be liminated.

TABLE 2. Web Service wDAG Similarity Results

Services	Similarity			Result
	Grounding	Model	Profile	
	1/4	1/4	2/4	
A111 - A011	0.9988	1.0000	0.9994	0.9994
A111 - A010	0.9988	0.9771	0.9734	0.9807
A111 - A110	1.0000	0.9771	0.9204	0.9545
A111 - A101	0.9450	0.8443	0.7817	0.8382
A111 - A001	0.9234	0.8443	0.7817	0.8328
A111 - A100	0.9205	0.7634	0.6650	0.7535
A111 - A000	0.8869	0.7634	0.6650	0.7451

We give weight wDAGsim [Profile] more than the others because it has a operator element that is very impotant in Web service similarity approach that we propose. From the results in the table above, the most appropriate Web service Web service replaces the "A111" is a Web Service "A011". This is due to it has same operator and profile with Web service "A111. Web service "A010" ranked second, this is becaused that it has the same service model with Web service "A111", although its service profile is different. This can be proven that a standardized description can be obtained from an operation or method of Web service like described Fig. 1 and Fig. 2.

Finally, our approach is to improve the method that has been proposed by Riyanarto Sarno [19], which is described as in Table 4.

TABLE 3. wDAG Similarities of Web Service Data Sets

Services	Similarity Score
S111 - S111	1.000
S111 - S011	0.97
S111 - S101	0.97
S111 - S110	0.86
S111 - S001	0.86
S111 - S100	0.67
S111 - S010	0.61
S111 - S000	0.56

Results of the similarity calculation between Web service "S111" with Web service "S101" is greater than the calculation simalaritas between Web service "S111" with Web service "S110", this indicates that the service model is not used to establish standard descritptions, so that Web service that different names and have the same model to replace the priority must not be a web service that needs to be replaced. So we get the wrong Web service to replace another Web service.

V. CONCLUSION AND FUTURE WORK

In this research, we have done some experiments about standarized descriptions. We found that the use of standardized descriptions is very important to find a replacement for proper web service. In the next research, we will use a web services available on the internet, calculate many formulas having same context but they are writen differently, and perform a composition among Web Services.

REFERENCES

[1] "Innovative design Solutions for the web", www.webservices.org, 2015

[2] H. Talantikite, D. Aissani, N. Boudjlida, S. Chabli et al., "The impact of web Services companies", Procedings of International Conference of Human Aspect in Advenced Manufacturing and Hybrid Automation, Manufacturing Enterprize of the future "HAAMAHA", Poznan, Poland, 2007.

[3] H. N. Talantikite, D. Aissani, N. Boudjlida, S. Aissani, et al., "Controle d'acce's aux web services complexes", Procedings of The International Conference of Information Systems and Economic Intelligence "SIIE", Tunisia, 2008.

[4] H. N. Talantikite, D. Aissani, N. Boudjlida, "Semantic annotation for web service descovery and composition", Computer Standarts and Interfaces 31, 2009, 1108-1117.

[5] J.E. Hadad, M. Manouvrier, M. Rukoz, "TQoS: transactional and QoS-aware selection algorithm for automatic web service composition", IEEE Trans. Serv. Comput. 3 (January–March) (2010) 73–85.

[6] J. Domingue, L. Cabral, H. Hakimpour, D. Sell, et al., "Irs-iii: a platform and infrastructure for creating wsmo-based semantic web services", Proceedings of the Workshop on WSMO Implementations (WIW), vol. 113 of 1613-0073, 2004, Frankfurt-Germany.www.wsmo.org/.

[7] "ESSI WSML working group", http://www.wsmo.org/wsml/, 2015.

[8] W3C, "Semantic Annotation for WSDL and XML Schema", www.w3.org/TR/2007/REC-sawsdl-20070828, 2007.

[9] R.Akkiraju, J. Farell, J. Miller, M. Nagarajan, et al, "Web Service Semantics-WSDL-S,UGA-IBM Technical Note", www.w3.org/2005/04/FSWS/Submissions/17/WSDLS.htm, 2005.

[10] W3C, " Semantic Annotations for WSDL and XML Schema", http://www.w3.org/TR/sawsdl/, 2015.

[11] Dengping Wei, Ting Wang, Ji Wang, and Abraham Bernstein, "SAWSDL-iMatcher: A customizable and effectiservice ve Semantic Web Service matchmaker", Web Semantics: Science, Services and Agents on the World Wide Web 9 (2011) 402–417.

[12] ChengZhi Xu et.al, " Semantic Web Services Annotation and Composition based on ER Model", IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing, 2010.

[13] Fatima-Zahra Belouadha, Hajar Omrana and Ounsa Roudiès, "A model-driven approach for composing SAWSDL semantic Web services", IJCSI International Journal of Computer Science Issues, Vol. 7, Issue 2, No 1, March 2010.

[14] Sonsilphong, S.; Arch-int, N.; Arch-Int, S., "Rule-based semantic web services annotation for healthcare information integration," Computing and Networking Technology (ICCNT), 2012 8th International Conference on , vol., no., pp.147,152, 27-29 Aug. 2012

[15] W3C, "OWL-S:Semantic Markup for Web Services", http://www.w3.org/Submission/OWL-S/, 2015.

[16] Jin, J., " Similarity of weighted directed acyclic graph" , Master's thesis, University of New Brunswick, September, 2006.

[17] G. Salton and C. Buckley, "Term-weighting Approaches in Automatic Text Retrieval," Information Processing and Management, vol.24, no.5, 1988, pp.513−523.

[18] Palmer Martha, and Zhibiao Wu, "Verb semantic and lexical selection", Proceding of the 32 th annual meeting on Assisiation for Computational Linguistic, Las Cruces, New Mexico, Assisiation for Computational Linguistic 1994.133-138.

[19] Riyanarto Sarno, Khakim Ghozali, Bayu Adhi Nugroho, and Astria Hijriani, "Semantic Matchmaking using Weighted Directed Acyclic Graph", Surabaya, 3rd Applied Technology Science and Art, 1-6.