Journal of ICT, Design, Engineering and Technological Science (JITDETS)
VOL. 1, NO. 2, pp. 42-47, 2017
DOI: https://doi.org/10.33150/JITDETS-1.2.3

ORIGINAL CONTRIBUTION

A Semantic Web Service Retrieval Approach that Combines Semantic Matching With Quality of Service Matching

Pornthip Liewtrakul [1*], Katawut Kaewbanjong [2]

[1] Faculty of Science & Technology, Bansomdejchaopraya Rajabhat University, Bangkok, Thailand

[2] Faculty of Business Administration and Liberal Art, Rajamangala University of Technology, Lanna Nan, Thailand.

Abstract— **The study objective is to propose a new web service search system that exploits the information structure of OWLS documents which consists of information from service history class, service model class, and basic service class. This information is used to construct indexes and their individual weights, which are used to compare the level of similarity between a semantic query from a user and each web service by a vector space model. Then, the result is used in a web service selection procedure based on formal concept analysis. A web service is selected through the structure of a concept lattice of Qos consisting of service availability and response time. Experimental results show that this proposed system provided an average accuracy of 71.9%. Moreover, it can provide alternative web services that are closely related to the query for the user to have flexibility in utilizing the search results.**

Index Terms— **Semantic Web Service, Quality of Service, Web Service Retrieval**

Received: 17 October 2017; **Accepted:** 8 November 2017; **Published:** 24 December 2017

I. INTRODUCTION

Service Oriented Architecture (SOA) is implemented by technology web service. Its advantages are reusability, loose coupling, platform independent, and interoperability. These advantages attract the attention of the private and educational sector, resulting in development of their applications into web services. A web service has 3 main components: Web Service Description Language (WSDL), Simple Object Access Protocol (SOAP) [1, 2, 3, 4], and Universal Discovery Description Integration (UDDI) [5] which can be described as follows.

WSDL is a document describing the capabilities of the service. It consists of functional description of the service operation element or service interface element based on XML language.

SOAP is a standard protocol for system connection. UDDI is a registry that stores the public business information of the service such as service name, business contact information, and location of the service. Please note that UDDI does not support functional search because it does not store the capability description of the service.

Due to the limitation of UDDI mentioned above and that of WSDL [6, 7] which is XML-based without any explanation of the capabilities of the service in a clear, meaningful way, they cause ambiguity or confusion in the interpretation of various service descriptions whether they are similar or different: for example, two web services may have the same service operation but different service descriptions, i.e., the explanations are not the same. For this reason, semantics is added to service description as a way to solve the problem. It has been developed from RDF to RDF-S, then to OWL and OWL-S in that order.

There have been several papers on solving the web service search problem [8, 9, 10, 11]. In this paper, we propose web service search based on exploration of OWL-S contents in the service profile component consisting of service name, text description, Input, Output, Precondition, and Effect which are elements used to construct the indexes for the service and the request. This study supports natural human language request. On top of that, we constructed a data structure that supports the search and is used in the determination of similarity between services according to the mentioned elements. We call this step Conceptual Indexing which will be explained in details in Section 4. Methodology.

The next section, Section 2, presents the background of problems of and solutions to web service retrieval. Section 3 briefly describes OWL-S specifications and data structure construction that supports web service search. Section 5 presents the experimental results and discussion. Section 6 is the conclusion of the paper.

II. RELATED WORK

Determination of similarity between web services according to web service concepts has been widely investigated. This section presents a brief survey of works pertaining to semantic web service discovery and matching of web services. The two main topics to be presented are the following. First, classification of web services by the content in the service description in either WSDL or OWL-S documents such as service name, operation, input, or output. Second, service matching by semantic similarity comparison between a service and a user query or between a service and another service.

[*] Corresponding author: Pornthip Liewtrakul
[†] Email: pliewtrakul@hotmail.com

In a previous study [8], information was extracted from WSDL document (operation name, input message, and output message) and a concept lattice was constructed from a context table that consisted of services for the purpose of clustering the services that are similar by consideration of the 3 pieces of information mentioned above together. In another study [9, 12, 13], clustering of services was done by consideration of semantic similarity between operation names. The services that passed the consideration criteria were used to construct a concept lattice.

In addition, [10] presents information extraction from WSDL that covered more features of WSDL than [8] and [9] did: content, type, message, port, and service name. Services with similar functionality were grouped according to the 5 features. All of these studies performed clustering of services with similar operation before constructing a concept lattice in order to reduce the size of the search space of the lattice.

All of the studies mentioned above are different than [14] in the sense that in [14], only the weights of the terms derived from the extracted information from WSDL document (operation name and parameter name) that passed specified criteria were used to construct a concept lattice that consisted of service and operation features. In [11], certain content of services was explored: Input, Output, Precondition, and Effect (IOPE). Semantic similarity distance and availability condition of services were considered. This consideration process was a preprocessing process of the model proposed in the study. It was done before a classification process by an FCA technique that required that the services and qos-level are included in the context table and led to selection of the service that was nearest to the query concept. This exploitation of concept lattice was different than that reported in [15]. In [15], services were classified by an analysis of 3 core components of OWL-S document: service profile, service grounding, and service model. The services were clustered according to the semantic similarity between operations for the purpose of constructing a concept lattice that were consisted of services and operations with an aim to find a substitution service when a master service was unavailable.

Semantic similarity matching analyzes the semantics of web service description content in WSDL and OWL-S documents. In [16], matching between query and service is presented. It considers the semantic similarity of precondition and effect which are features of OWL-S document. The matching process consists of 3 phases: parameter compatibility, condition equivalence, and condition evaluation. The outcome can be one of these 4 types: exact, plugin, subsume, or disjoin. The paper [17] presents a multi-dimensional matching ontology or MDM-Onto. It considers the semantic correspondence between input/output parameters and QoS. The paper [18] presents a full utilization of terms in WSDL document. The terms were used to measure the similarity between services with respect to semantic distance and external knowledge (WordNet).

From this survey of literature, it was found that IR metric or tf-idf from exploration of the content in service description was only used for filtering the indexes but not applied to service matching. Also, matching is still not flexible enough to support human language query. Most studies deal only with a specific form of query such as input type or operation type. Moreover, classification of semantic web service documents by an FCA technique has still received little attention, as reported in [11] and [18].

In the next section, details of OWL-S document are discussed briefly. These details are service descriptions that were used in this study. Additionally, it discusses FCA, the classification technique used in this study.

III. METHODOLOGY

The design of a model for semantic web service retrieval consisted of 3 main steps: preprocessing, semantic matching, and quality of service matching. The design framework is shown in Figure **??** below.

A. Preprocessing Step

In this study, data from semantic web services were analyzed. The data were represented by OWLS language' structure which consisted of service profile, service model class, a service grounding class. The analysis consisted of various steps shown in Figure 1: 1) detail extraction, 2) web service index construction, and 3) calculation of index weights.

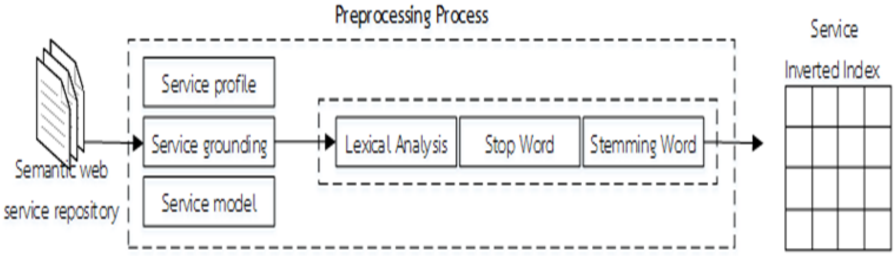
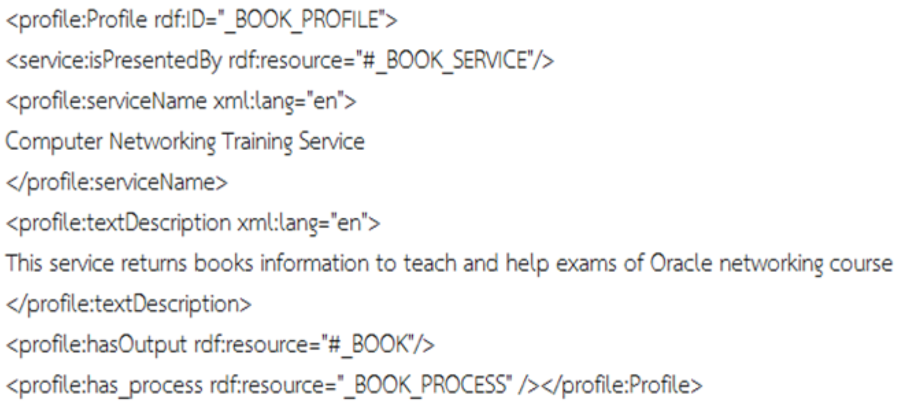


Fig. 1. Preprocessing steps

B. Detail Extraction

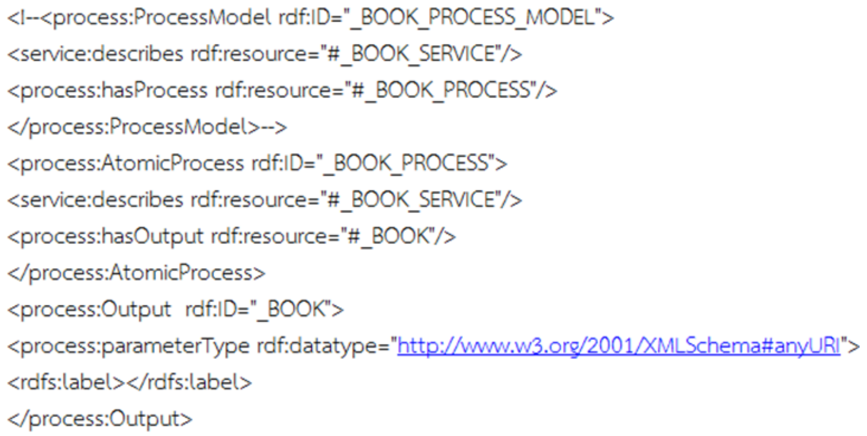
Details were extracted from a service profile class for construction of an index for web service retrieval. The details were 2 properties and 4 functions as follows: 1) Service name, 2) Text description about web construction, 3) Has output which is an explanation of data export that is the processed result of specifying the import data, 4) Has input which is an explanation of data import which will be processed according to the type of data specified, 5) Has precondition which is an explanation of the necessary conditions that need to be verified and processed before data import and 6) Has effect which is the processed result that depends on the information in the Has precondition.



```
<profile:Profile rdf:ID="_BOOK_PROFILE">
<service:isPresentedBy rdf:resource="#_BOOK_SERVICE"/>
<profile:serviceName xml:lang="en">
Computer Networking Training Service
</profile:serviceName>
<profile:textDescription xml:lang="en">
This service returns books information to teach and help exams of Oracle networking course
</profile:textDescription>
<profile:hasOutput rdf:resource="#_BOOK"/>
<profile:has_process rdf:resource="_BOOK_PROCESS" /></profile:Profile>
```

Fig. 2. Components of the service profile class of Book Oracle Service. Owls web service [10]

As can be seen in Figure 3, details were extracted from a service model class.of the model and used them as indexes for web service retrieval. These details can be considered as the following 4 functions. Has precondition is an explanation of the necessary operation and process for verifying the service conditions; has input is an explanation of data import for processing according to the quality or quantity of the import data. Has output is an explanation of data export for processing according to the quality or quantity of the export data.Has effect is the processed results of the operation on the data that are included in the Has precondition.



```
<!--<process:ProcessModel rdf:ID="_BOOK_PROCESS_MODEL">
<service:describes rdf:resource="#_BOOK_SERVICE"/>
<process:hasProcess rdf:resource="#_BOOK_PROCESS"/>
</process:ProcessModel>-->
<process:AtomicProcess rdf:ID="_BOOK_PROCESS">
<service:describes rdf:resource="#_BOOK_SERVICE"/>
<process:hasOutput rdf:resource="#_BOOK"/>
</process:AtomicProcess>
<process:Output  rdf:ID="_BOOK">
<process:parameterType rdf:datatype="http://www.w3.org/2001/XMLSchema#anyURI">
<rdfs:label></rdfs:label>
</process:Output>
```

Fig. 3. Components of data from the service model class

C. Web Service Index Construction

For an information retrieval system, indexes facilitate searches for information greatly. The preliminary extraction steps mentioned above may result in data that cannot be used for index construction. Therefore, these data have to be pre-processed. Preprocessing consists of lexical analysis, finding stop word, and finding the root word [15]. Finding stop word is a process of elimination of high frequency words. Words of this type, such as preposition and punctuation, lack identifying capability and so they are useless as an index.Finding stemming word is a process of combining words that come from the same root together in order to reduce the size of the indexes; for example, 'computer', 'computing', and 'computed' are combined together into a group that is represented by a word in the group. After the document was pre-processed, index weights are calculated for using as indicators of the level of similarity of the document to the query.

1) Calculation of index weights: This step assigns a weight to each word that represents the semantic web service document.In general, a weight is calculated by multiplying Term Frequency (TF) by Inverse Document Frequency (IDF), where;

$$idf_k = log(N/n_k)(1) \tag{1}$$

Web service index weight can be calculated by the equation below;

$$W_{i,k} = tf_k * idf_k(2) \tag{2}$$

where $w_{i,k}$ is the weight of the kth index in the ith web service and tf_k is the frequency of the kth index in the ith web service, where $i \in (1...N)$ and $k \in (1...M)$.

TABLE I
SOME INDEX WEIGHTS

ID	Terms	IDF	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11	S12	S13	S14	S15	S16	S17	S18	S19	S20
1	Visit	2.13672057																				
2	Research	0.99059253																				
3	Address	1.02277721									5.11	5.11										
4	Home	1.43775956									1.44											
5	Office	1.65959931																				
6	Senior	1.83569057																				
7	Lecturer	1.09532788					5.48	5.477	5.5													
8	University	0.88144806					1.76	1.763	2.6		0.88	0.88										
9	Professor	1.65959931																				
10	Name	0.88144806					0.88	0.881	0.9													
11	Teach	1.65959931			1.66																	
12	Semester	2.13672057																				
13	Current	1.8369057	1.8																			

D. Semantic Matching

Vector comparison between web service document vector and query vector can be done by similarity calculation between the semantic web service document and the query vector [19] which can be done by determining the dot product between the vectors. The result is a numerical indicator of similarity in the range of [0, 1], calculated by Equation 3 below;

$$Sim(q, d_i) = \frac{\sum_{k=1}^{N} (w_{q,k}) * (W_{i,k})}{\sqrt{\sum_{k=1}^{N} (W_{q,k})^2 * \sum_{k=1}^{N} (W_{i,k})(W_{i,k})^2}} \quad (3)$$

where $sim(q, d_i)$ is the level of similarity between the document and the query, ($w_{i,k}$)is the weight of the kth index in the ith document, where *i* ϵ 1...N, *N* is the total number of documents, and ($w_{q,k}$) is the weight of the kth index in the query. Suppose a query consists of 3 keywords: German, government, and scholarship. By Equation 3, values of similarity between the query and every web service in the example (20 services, S_1-S_{20},) can be calculated as follows.

$Sim(q, s_{13})$ = (2.584)/(1.398*6.799) = 0.271
$Sim(q, s_{14})$ = (2.584)/(1.398*7.026) = 0.262
$Sim(q, s_{15})$ = (2.584)/(1.398*5.585) = 0.330
$Sim(q, s_{16})$ = (2.584)/(1.398*7.473) = 0.247
$Sim(q, s_{17})$ = (4.574)/(1.398*5.081) = 0.643
$Sim(q, s_{18})$ = (5.237)/(1.398*6.706) = 0.558
$Sim(q, s_{19})$ = (4.479)/(1.398*4.479) = 0.105

Seven web services, s_{13}, s_{14}, s_{15}, s_{16}, s_{17}, s_{18}, s_{19}, match well with the query having corresponding similarity values in the range of 0.105 - 0.643.

E. Quality of Service Matching

This step is just like the determination of similarity of web services in Step 2 but queried with a quality of service query (qosQuery) consisting of 2 features: Availability (Av) and Response Time (RT). This determination consists of the following steps: 1) specification of levels of quality of service; 2) context Table I construction; and 3) concept lattice construction and search. An illustrated diagram of all of the steps is shown in Figure 5.

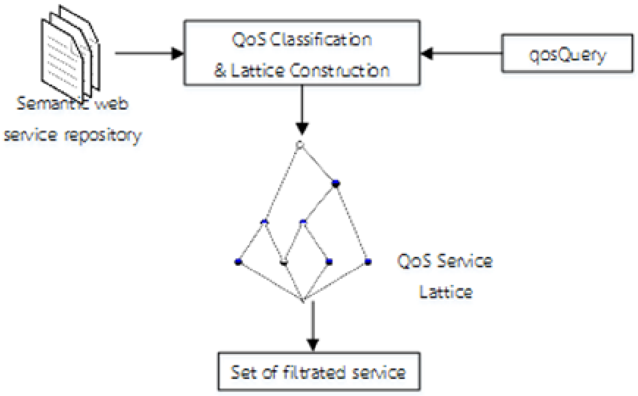
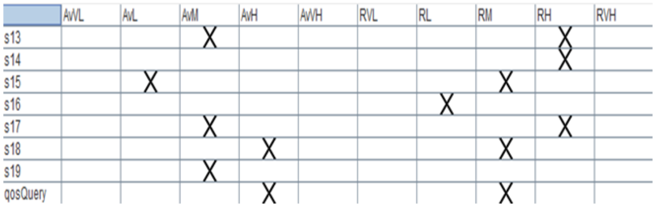


Fig. 4. Procedural steps in the determination of similarity between web services and quality of service query

1) Specification of levels of quality of service: Levels of Av and Rt are specified as 5 class intervals: Very Low (VL), Low (L), Medium (M), High (H), and Very High (VH).

2) Context Table I construction: In this step, a context Table I is constructed to consist of semantic relationships between web services: Si = s13, s14, s15, s16, s17, s18, s19; qosAttribute= Av, Rt; and qosLevel = VL, L, M, H, VH, represented byc= (S$_i$, qosAttribute, qosLevel) as shown in Figure 6.



	AVVL	AvL	AvM	AvH	AVVH	RVL	RL	RM	RH	RVH
s13			X						X	
s14									X	
s15		X						X		
s16							X			
s17			X						X	
s18				X				X		
s19			X							
qosQuery				X				X		

Fig. 5. A context Table I of c = (S$_i$, qosAttribute, qosLevel

3) Concept lattice construction and web service search: The context Table I of the relationship c = (Si, qosAttribute, qosLevel) is used to construct a concept lattice, with a program called 'Concept explore', shown in Figure 6.

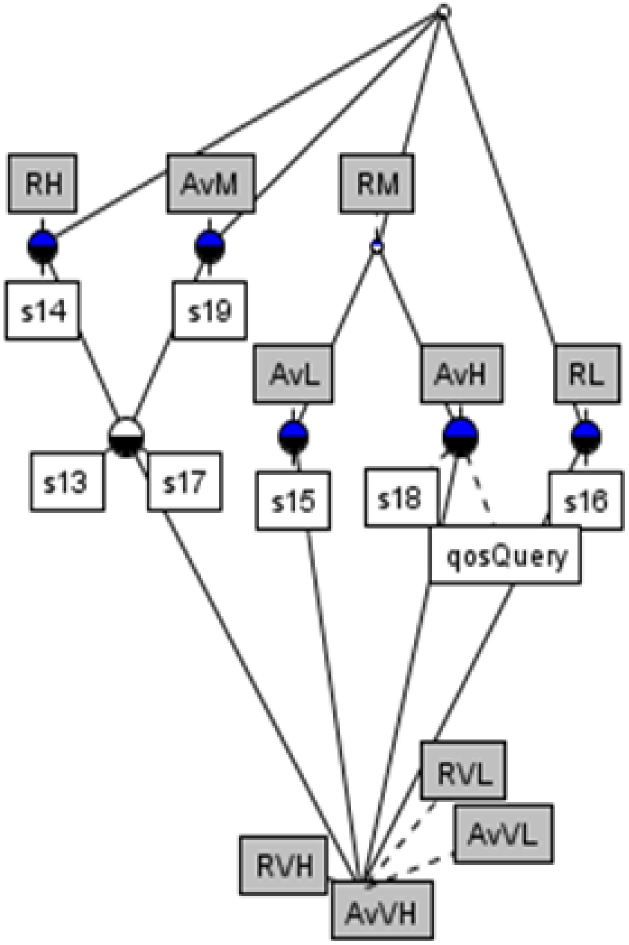


Fig. 6. A concept lattice constructed from the context Table I of c = (S_i, Av-level

The semantic web service search from concept lattice structure employed in this study add qosQuery to the relationship c resulting in c =(Si, qosAttributes, qosLevel, qosQuery) which includes qosQuery as a node in the lattice. Hence, a search is qosQuery-navigated. The similarity between qosQuery and web services in the concept lattice structure is specified [19] as follows:

Exact match means that qosQuery and qos web service advertisement are conceptually equivalent, Subsume match means that qosQuery is a super concept of qos web service advertisement; Plugin match means that qosQuery is a sub-concept of qos web service advertisement.

Partial match means that qosQuery overlaps with qos web service advertisement and they are not equal. Fail match means that qosQuery is unrelated to qos web service advertisement. Suppose that qosQuery = High Availability, Medium Response time, as can be seen in Figure 6, the semantic web service search for a service that is similar to the query is done in the following way: System Recommend = S_{18} is an exact match (High availability, Medium response time); Alternative Recommend = S_{15} is a partial match (Low availability, Medium response time).

IV. CONCLUSION AND DISCUSSION

This study proposed a semantic web service retrieval system whose procedure consists of 3 main steps: index construction for semantic searching from the structure of web service documents, determination of semantic similarity by a vector space model and using it to construct a concept lattice for search with a quality of service query consisting of two features: availability and response time. The output is a web service that is semantically similar to the user's query and of a level of service quality that the user requests. The performance of this retrieval system was tested with data from OWLS-TC v. 4. It was used in the test as test documents in combination with 100 queries from 10 students who already had some basic knowledge about web service. The author acted as an expert on the analysis of suitability of search result. The results showed that this retrieval system was able to find good matches from keywords at 71.9% accuracy.

In addition, several pieces of useful information were acquired in the course of this study as follows: 1) Index construction from every feature of an OWLS document (service history class, service format class, and basic service class) expands the range of keywords as well as the frequencies of the keywords. Nevertheless, some keywords are a single word with a specific meaning which may have a lot of similar words, so it was better to semantically classify the keywords again by comparing them to those in the Word Net dictionary. 2) In addition, in the determination of similarity between web services and the query with a vector space model, this study only used the web services with similarity values in the ranges of [0, 1] and [0.5, 1] in the construction of concept lattice later on, and it was found that a specification of this range had a direct relationship with the number of web services that were going to be used to construct the context Table I and with the amount of time to construct the concept lattice. 3) and lastly, a limitation of the kind of search from a concept lattice used in this study is that it may result in too large a number of web services with common attributes from partial matching which may cause a user some difficulty to decide and select one since there is no ranking criteria at this present stage of development. This project will continue to develop in the future.

Declaration of Competing Interest

The authors declare that they have no conflict of interest.

References

[1] Web Service Description Language (WSDL), ``Latest soap versions,'' 2007. [Online]. Available: https://bit.ly/2PQiZuL

[2] Q. A. Le, ``The impact of globalisation on the reform of higher education in Vietnam,'' *International Journal of Business and Economic Affairs*, vol. 1, no. 1, pp. 29-35, 2016. doi: https://doi.org/10.24088/ijbea-2016-11005

[3] M. El Amrousi and F. I. Abdulhafiz, ``Applications of technology in design disciplines: Perforated screens and ornament, the impact of digitization in higher education,'' *International Journal of Technology and Engineering Studies*, vol. 3, no. 6, pp. 236-244, 2017. doi: https://doi.org/10.20469/ijtes.3.40003-6

[4] J. D. Febro and J. Barbosa, ``Mining student at risk in higher education using predictive models,'' *Journal of Advances in Technology and Engineering Research*, vol. 4, no. 1, pp. 17-132, 2017. doi: https://doi.org/10.20474/jater-3.4.2

[5] Universsal Description Discovery and Integration, ``UDDI technical white paper,'' 2000. [Online]. Available: https://bit.ly/2QDM4Pk

[6] Web Service Description Language (WSDL), ``Resource Description Framework (RDF): Concepts and abstract syntax,'' 2004. [Online]. Available: https://bit.ly/2PMQSMZ

[7] V. Codocedo, I. Lykourentzou, and A. Napoli, ``Semantic querying of data guided by formal concept analysis,'' in *The International Confer-*

ence on Formal Concept Analysis for Artificial Intelligence, Montpellier, France, 2012, p. 23.

[8] D. Peng, S. Huang, X. Wang, and A. Zhou, ``Management and retrieval of web services based on formal concept analysis,'' in *The Proceedings of the fifth International Conference on Computer and Information Technology,* Shanghai, China, 2005.

[9] D. L. Balasubramanian, S. Murugaiyan, G. Sambasivam, T. Vengattaraman, and P. Dhavachelvan, ``Semantic web service clustering using concept lattice: Multi agent based approach,'' *International Journal of Engineering and Technology*, vol. 5, no. 5, pp. 3699-3714, 2013.

[10] K. Elgazzar, A. E. Hassan, and P. Martin, ``Clustering wsdl documents to bootstrap the discovery of web services,'' in *The Proceedings of the IEEE International Conference on Web Services,* Miami, FL, 2010.

[11] X. Jianhong, G. Weizhi, and Y. Wang, ``A cloud service discovery approach based on fca,'' in *2nd International Conference on Cloud Computing and Intelligence Systems,* Hangzhou, China, 2012.

[12] C. L. S. Tablatin, F. F. Patacsil, and P. V. Cenas, ``Design and development of an information technology fundamentals multimedia courseware for dynamic learning environment,'' *Journal of Advances in Technology and Engineering Studies*, vol. 2, no. 5, pp. 202-210, 2016.

[13] T. Limgomonvilas, ``Prediction for nonthaburi urban parks by integrated geo-informatics techniques,'' *International Journal of Technology and Engineering Studies*, vol. 3, no. 1, pp. 20-28,, 2017. doi: https://doi.org/10.20469/ijtes.3.40003-1

[14] L. Aversano, M. Bruno, G. Canfora, M. Di Penta, and D. Distante, ``Using concept lattices to support service selection,'' *International Journal of Web Services Research (IJWSR)*, vol. 3, no. 4, pp. 32-51, 2006. doi: https://doi.org/10.4018/jwsr.2006100102

[15] A. Abid, N. Messai, M. Rouached, T. Devogele, and M. Abid, ``A semantic similarity measure for conceptual web services classification,'' in *24th International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises,* Larnaca, Cyprus, 2015.

[16] U. Bellur and H. Vadodaria, ``On extending semantic matchmaking to include preconditions and effects,'' in *IEEE International Conference on Web Services,* Beijing, China, 2008.

[17] L. Guzman, M. Regina, R. Alvaro, and S. Marco, ``Multidimensional semantic web services matching,'' in *The Proceedings of the International Conference on LA Web,* Buzios, Brazil, 2016.

[18] F. Liu, Y. Shi, J. Yu, T. Wang, and J. Wu, ``Measuring similarity of web services based on wsdl,'' in *International Conference on Web Services,* Miami, FL, 2010.

[19] P. Pukkasenung, P. Sophatsathit, and C. Lursinsap, ``An efficient semantic web service discovery using hybrid matching,'' in *Proceedings of the 2nd International Conference on Knowledge and Smart Technologies,* Rabat, Morocco, 2010.