**International Journal of Electrical and Computer Engineering (IJECE)**
Vol. 9, No. 2, April 2019, pp. 1437~1444
ISSN: 2088-8708, DOI: 10.11591/ijece.v9i2.pp1437-1444 

# A new approach to gather similar operations extracted from web services

**Rekkal Sara, Amrane Bakhta, Loukil Lakhdar**
Department of Computer Science, University of Oran 1 Ahmed Ben Bella, Algeria

**Article Info**

***Article history:***

Received Jun 9, 2018
Revised Nov 9, 2018
Accepted Nov 24, 2018

***Keywords:***

Hungarian algorithm
Semantic analysis
Syntax analysis
Web services
WSDL similarity

**ABSTRACT**

A web service is an autonomous software that exposes a set of features on the Internet, it is developed and published by providers and accessed by customers who discover it, select it, invoke and use it. Several research policies have been implemented such as searching through keywords, searching according to semantics and searching by estimating the similarity. A customer is looking for a service for the operations he/she carries out, hence the interest of guiding the search for services towards a search for operations: finding the desired operations amounts to finding the services. For this, groupings of similar operations would make it possible to obtain all the services that can meet the desired functionalities. The customer can then select, in this set the service or services according to its non-functional criteria. The paper presents a study of the similarity between operations. The proposed approach is validated through an experimental study conducted on web services belonging to various domains.



***Corresponding Author:***

Rekkal Sara,
Department of computer science,
University of Oran 1 Ahmed Ben Bella, Oran, Algeria.
Email: rekkal_sarah@yahoo.fr

## 1. INTRODUCTION

Developed by providers in WSDLs (Web Service Description Language) files and exposed in UDDI (Universal Description, Discovery, and Integration), web services are programs that define features with a view to being remotely called upon through an SOA architecture (Service-Oriented Architecture). For various reasons, the main ones being their availability and easy-to-use, web services have quickly gained popularity and their number keeps growing.

The manipulation process begins with the interrogation of the UDDIs registers with a simple request that expresses the client's need. For a given need, several Web services may exist, each with particular characteristics such as cost, performance, reliability etc. The client can then select the service that best meets his needs, retrieves the URL identifying the service of his choice, access his WSDL interface and proceed to its invocation.

Despite the efforts of research and development around Web services, these technologies are extremely complex and pose many challenges. Their complexities usually come from the following sources:

1. The marketing of web services on the internet is constantly increasing, which results in an increasingly growing number:
   a. Making research difficult and greedy in terms of time.
   b. Leading to an insufficient selection of relevant services.
2. Web services are created and updated in a highly dynamic way. The same services today can be different tomorrow.

3. Web services are developed by different entities. As a result, for the same task, we can discover several Web services capable of executing it. However, they differ in their non-functional properties which are expressed by attributes called QoS "Quality of Service".
4. Web services operate in volatile environments. As a result, their relocation or deletion is done on the fly, which causes the applications to stop when they are used.

Many works have been proposed to address those problems such that:

In [1] the authors suggested an approach in order to determine the similarity between web services using their WSDLs descriptions. First of all, they reduce the service to a set of operations, on which they implemented a method that enabled to reorganize their descriptions within the WSDL file in order to apply NiCad clone detector in order to identify fragments that are very similar. The NiCad clone detector uses an efficient and scalable hybrid parsing and text comparison technique based on the Longest Common Subsequence (LCS) algorithm to identify near-miss clones. WSDL is not a source code, so two operations may appear similar when in fact they are not.

Doug et al in [2] suggested a clustering algorithm that gathers together parameters names into a meaningful concept, they use the following heuristic: parameters that often appear together tend to express the same thing; this algorithm was implemented in Woogle which is a search engine for web services. Clustering of the identifiers obtained neglects the types used.

Authors of [3] suggested a technique for lexical and structural similarity assessment of web service descriptions; their similarity study is based on the measurement of similarity between descriptions (documentation) of various elements. Usually, these descriptions do not always appear in WSDLs files.

Rashad et al in [4] proposed an approach for measuring the similarity between the identifiers of operations based on a semantic and syntactic study. The study ignores different types that have a part in the result of the study.

The work in [4] allowed the authors in [5] the adoption of a method for the substitution of web services. A work of detecting correspondence relations between operations was realized (For two similar Web services, an operation of one replaces an operation of the other). However, some relationships do not conform to reality: the relationship defined by different inputs and identical outputs does not allow the substitution because the operations concerned by these parameters are considered different. Also some relations defined in [5] are not significant, as example the relation of corestriction between operations.

As part of this work, we focus first of all on the reorganization of the web services space by forming clusters (Services, Common Operations), Figure 1. These communities:

a. Facilitate the search task while reducing the search time.
b. Provide all relevant operations.
c. Ensure high availability of Web service operations (substitution).

The paper is organized as follows: Section 2 is devoted to the presentation of related works; Section 3 describes the proposed approach. In Section 4 we report and analyze the results of the experiments. Finally, we end with a conclusion.

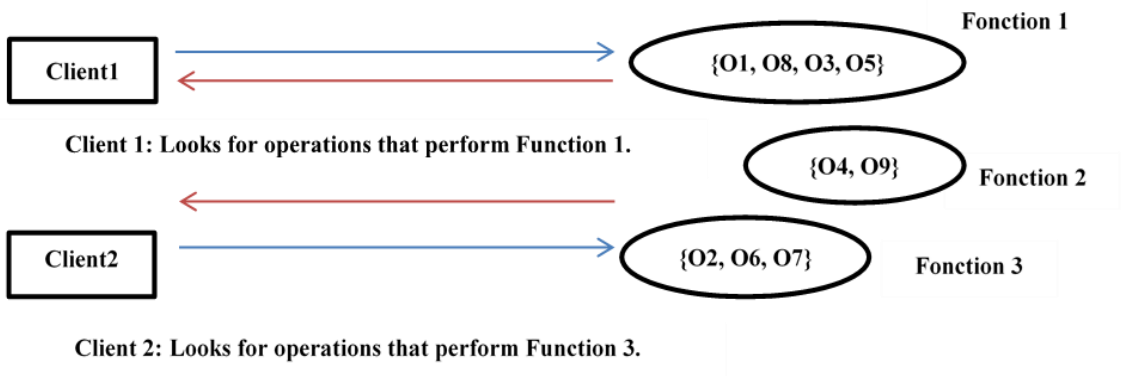


Figure 1. Research in an organized space

## 2. RELATED WORK

Using different methods and techniques, many works have focused their studies on WSDLs files, SOA architecture, Internet of thing, etc. In [6] the authors, to measure the similarity of web services, they

implemented three functions that successively return a similarity value between the web services' identifiers, a similarity value between their operations and a similarity value between their descriptions and that by exploiting at the same time semantic similarity measurements and syntactic ones. The validation of the results obtained is totally absent.

The authors of [7] present an IoT Crowd Sensing platform that offers a set of services to citizens by exploiting a network of bicycles as IoT probes. Based on a survey conducted to identify the most interesting bike-enabled services, the SmartBike platform provides: real time remote geo-location of users' bikes, anti-theft service, information about traveled route, and air pollution monitoring. The proposed SmartBike platform is composed of three main components: the SmartBike mobile sensors for data collection installed on the bicycle; the end-user devices implementing the user interface for geo-location and anti-theft; and the SmartBike central servers for storing and processing detected data and providing a web interface for data visualization. The suitability of the platform was evaluated through the implementation of an initial prototype. Results demonstrate that the proposed SmartBike platform is able to provide the stated services, and, in addition, that the accuracy of the acquired air quality measurements is compatible with the one provided by the official environmental monitoring system of the city of Turin. The described platform will be adopted within a project promoted by the city of Turin that aims at helping people making their mobility behavior more sustainable.

In [8] the author describes the use of Service Oriented Architecture (SOA) architecture with Web services architecture to accurately solve techniques that have a lower negative impact in terms of performance and service security.

The authors of [9] proposed a face recognition security system using Raspberry Pi which can be connected to the smart home system. Eigenface was used the feature extraction, while Principal Component Analysis (PCA) was used as the classifier. The output of face recognition algorithm is then connected to the relay circuit, in which it will lock or unlock the magnetic lock placed at the door. Results showed the effectiveness of our proposed system, in which we obtain around 90% face recognition accuracy. We also proposed a hierarchical image processing approach to reduce the training or testing time while improving the recognition accuracy.

The work done in [10] discusses the existing routing trend in IoT, vision and current challenges. This paper also elaborates the technologies and domains to drive this field for future perspectives. The paper concludes with discussion and main points for new researchers in terms of routing to understand about current situation in Iot.

In [11] authors shows what the future of the Internet is, thier research carries out a qualitative prospective analysis on projects and investigations in which the scientific community is currently working, the information is analyzed, and the highlighted topics are shown. The principle aim of the work done in [12] is to aid the visually challenged human beings with a better navigation device. This clever strolling stick is greater state-of-the-art device with many embedded features.

In [13], three main points that are concentrated 1) Design a robot which is vehicle-mounted sensors that capable of carries the sensors of the metal and obstacle; 2) Control and management system wirelessly by a computer-based to command the robot functions by several sets of user's rules and manage the robot instructions; and 3) Conduct an integrated system that achieving navigated data via metal detector based on online structured query language database registry. Also, discussed a comparison of the previous detector systems and highlights on several merits. The proposed system capable of fully control the robot also, set the robot operator permissions and rules, stored and archived the navigated results and printed reports and stored in an independent database.

## 3. PROPOSED APPROACH

### 3.1. Syntactic and semantic analysis

Since it is difficult, even impossible to access their source codes, studying the similarity of web services operations consists in studying their descriptions extracted from WSDLs files. A similarity analysis, in this case, imposes methods of automatic processing of natural language. Two types of analysis can be considered: a syntactic and a semantic analysis. These return similarity measures between [0, 1], where 0 means dissimilar and 1 means similar.

a) Syntactic analysis is based on the similarity of the structures of the chains to be compared. Several methods exist of which we chose to use Jaro-Winkler because it is the best-known algorithm and according to [5] it is the most powerful and fastest measure.
b) Semantic analysis is based on the resemblance of the semantic level. Several methods exist, of which Wu-Palmer was chosen because it has the advantage of being simpler to implement and gives better results according to [5].

### 3.2. Hungarian maximum matching

The Hungarian method, or Kuhn-Munkres' algorithm, is an algorithm of combinatorial optimization that solves the assignment issue. It is, therefore, an algorithm that allows finding a perfect coupling of maximum weight in a bipartite graph. A graph can be represented by a matrix whose cells are considered to be the edges of the graph. A match is a subset of edges where two edges in the subset cannot share a common vertex. In other words, it is a set of values in the matrix where two values can never be in the same line or column.

### 3.3. Structure of WSDL FILE

A WSDL file is an XML file. It describes the functionality of a web service. Through this file, we have access to the following elements:

a) The name and the description of the web service.
b) The name and the description of the operations (containing the inputs and outputs messages).
c) The name and the description of the messages (inputs/outputs) and their associated parameters.
d) The name and the description of the parameters that can be of a simple or a complex type.

### 3.4. Similarity process between operations

In our work, the study of similarity between operations consists in:

a) Extracting the necessary elements from the WSDLs files.
b) Transforming complex parameters into simple parameters.
c) Evaluating the similarity between the parameters of the compared operations.
d) Determining the relationship between the compared operations.
e) Grouping similar operations.

#### 3.4.1. Extract the necessary elements from the WSDLS files

The elements to extract are:

a) Service Name: information that gives the source of the operation. (The X Operation is obtained from the Y service).
b) Operation Name: information that will constitute the clusters and that intervenes in the similarity study.
c) Outputs (message and parameters identifiers' and their associated types): information that intervenes in the similarity study.

#### 3.4.2. Transform complex parameters into simple parameters

The parameters of an operation can be simple or complex. A simple parameter is described by its name and type. A complex parameter describes a structure composed of different elements. The study of similarity on the latter requires its transformation into simple parameters by aggregation of the identifiers from the sub-elements with their parent.

#### 3.4.3. Evaluate the similarity between the parameters of the compared operations

a. Definition

Any operation takes a set of input parameters and produces a set of output parameters, whatever the relationship (intersection, difference, equality, or inclusion) that may connect the inputs of two given operations. A client is interested in the results they produce. The relationship between operations depends on the connection between their respective outputs. Hence, the idea of grouping operations that produce the same output (same results).

b. Similarity process

Let O1 and O2 be two operations extracted from different services S1 and S2. Let {parameters} and {parameters'} respectively be their output parameters. A parameter is defined by an identifier and a type that can be simple or complex.

In this work, each identifier will be concatenated with the identifier of the operation and that of the output message. The study of similarity consists of:

1. Building a matrix of similarity whose horizontal lines refer to the parameters of the first operation and the columns refer to the parameters of the second operation Table 3. The following formula determines the values of this matrix:

$$Sim\_Pars=(Sim\_ident\ () + Sim\_Types\ ())/2. \quad (1)$$

where:

(1) Sim_Pars (): is the main function, which calculates the similarity between parameters. This measure is included between 0 and 1.

(2) Sim_ident (): is the function that measures the similarity between the identifiers of the output parameters. This similarity measure consists of:
   a. Chopping identifiers into words.
   b. Remove stop words, as well as special characters and numbers.
   c. Extending the abbreviations.
   d. Lemmatizing the segments (use the singular, the infinitive for verbs, the masculine for adjectives, etc.).
   e. Building a similarity matrix between the words of two different identifiers, where columns represent the words of the first identifier and the lines those of the second, Table 1. The matrix values are determined according to a semantic analysis (WU-PALMER) if the two words exist in the Word-net, if not, by using a syntactic analysis (Jaro-Winkler).
   f. Calculate the degree of similarity between the identifiers, by calculating the average of maximum scores (the Hungarian method).

Table 1. Table of Similarity between Two Identifiers

| Identifier1 | Identifier2 | | |
|---|---|---|---|
| | Word1 | Word2 | Word3 |
| Word'1 | Wu-Palmer/ Jaro-Winkler | Wu-Palmer/ Jaro-Winkler | Wu-Palmer/ Jaro-Winkler |
| Word'2 | Wu-Palmer/ Jaro-Winkler | Wu-Palmer/ Jaro-Winkler | Wu-Palmer/ Jaro-Winkler |

(3) Sim_Types (): is the function that measures the similarity between the types of parameters. Let T be a type: integer, real, string, date or boolean. [14] and [15] propose Table 2, which determines the similarity between the different possible types:

Table 2. Similarity between Types [6]

| | Integer | Real | String | Date | Boolean |
|---|---|---|---|---|---|
| Integer | 1.0 | 0.5 | 0.3 | 0.1 | 0.1 |
| Real | 1.0 | 1.0 | 0.1 | 0.0 | 0.1 |
| String | 0.7 | 0.7 | 1.0 | 0.8 | 0.3 |
| Date | 0.1 | 0.0 | 0.1 | 1.0 | 0.0 |
| Boolean | 0.1 | 0.0 | 0.1 | 0.0 | 1.0 |

The similarity between two given types of parameters is calculated by the following formula:

$$Sim\_Types=\min(Sim(T1, T2), Sim(T2, T1)). \quad (2)$$

where:

T1 is the type of the first parameter and T2 is the type of the other.

2. Applying the Hungarian method on the similarity matrix of parameters in order to obtain the maximum scores (without calculating their averages) see Table 3.

Table 3. Similarity between Parameters

| O1 | O2 | | |
|---|---|---|---|
| | Parameter1 | Parameter2 | Parameter3 |
| Parameter'1 | Sim_Pars () | Sim_Pars () | Sim_Pars () |
| Parameter'2 | Sim_Pars () | Sim_Pars () | Sim_Pars () |

c. Threshold that determines similarity

It is clear that the more the value obtained tends to 1, the more the compared parameters are similar. From a threshold equal to 0.7 the parameters are considered similar.

### 3.4.4. Determining the relationship between the compared operations

Based on the maximum scores obtained from the previous step (similarity calculation), we can define two relationships between operations:

a. Similarity: If two operations have the same number of parameters and all similarity values obtained are>= threshold then operations are considered similar.
b. Similarity with excess: An operation O1 is similar with excess with O2 if the number of its parameters is higher and all the similarity values obtained are>= threshold, which means that the O2 functionalities can be realized by O1 even though the latter produces more parameters not requested.

### 3.4.5. Grouping similar operations

A grouping according to the determined similarity relation can be considered. Thus clusters of similar operations are constituted, connected by arcs to clusters grouping included operations. The orientation of the arc expresses the similarity with excess and its orientation indicates the direction of inclusion Figure 2. So, if the client requests an operation that can be satisfied by {O1, O6, O10}, {O2, O5} will be also returned; but this doesn't work the opposite way.


Figure 2. The result of the similarity study between operations

## 4. THE EXPERIMENT RESULTS

The experiment has been carried out on real web services belonging to different fields: communications, transport,finance, weather.

### 4.1. Implementation

The approach has been applied on an Intel processor machine (I3-3110M CPU2.40GHZ) with 4GB RAM and Windows 07 as the operating system.

### 4.2. Results and assessment

The tool has been experimented with 10, 24, 39 WSDLs samples. To verify the accuracy and efficiency of our approach, we performed a manual assessment of the similarity of the operations on the same sample (human evaluation), Since we do not have similar tools to compare their results against those generated by our tool. We have run two tests:

1) Test 01

We have measure precision and the recall, Figure 3, Figure 4 summerize the obtained results.

a. Precision measures the proportion of software results that are considered relevant or correct, and it is the ratio of the number of relevant items found by the total number of items found.
b. Recall measures the proportion of all the correct results that a software might theoretically find, and it is the ratio of the number of relevant elements found by the total number of relevant elements.

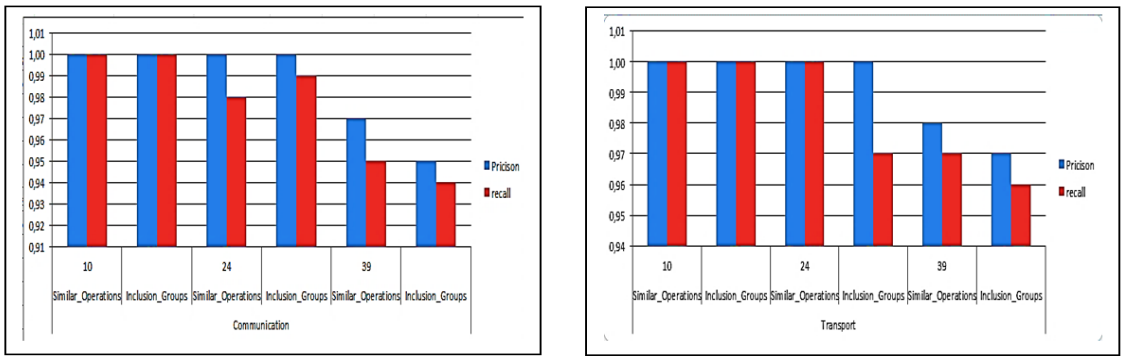


Figure 3. Precision and recall results for communication and transport WSDL files

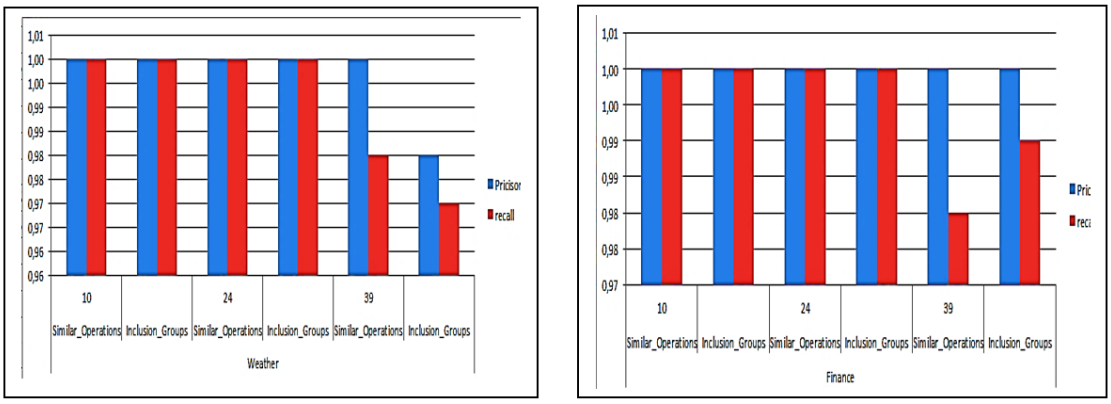


Figure 4. Precision and recall results for weather and finance WSDL files

The results obtained show that our approach is very efficient, it is simple and precise compared to other approaches which thheir process is very complex.

2) Test 02

We measured the similarity between two partitions (of one field of study: Weather): the calculated ones by our tool "C" with the correct partition obtained manually "P", using Rand's index, such that:

$$Rand\ (C,P)=\frac{a+d}{(a+b+c+d)} \quad (3)$$

where:

a) 'a': the number of pairs of co-grouped elements in both partitions.
b) 'd': is the number of non-co-grouped item pairs in either partition.
c) 'c': corresponds to the number of pairs of elements co-grouped in C but not in P.
d) 'b': corresponds to the number of pairs of elements co-grouped in P but not in C.

The obtained results are summarized in Figure 5. It is important to note that "C" and "P" include both groups of similar operations and groups of inclusion ones.The obtained results from the rand index confirm the homogenization of the formed groups.

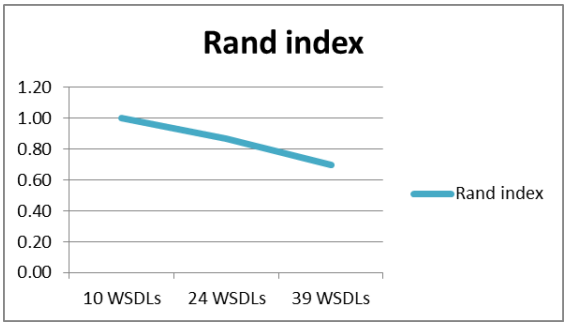


Figure 5. Rand index results

## 5. CONCLUSION

We have suggested a simple and effective approach to the study of similarity between operations of web services in order to make up groups of similar operations responding to the same needs, i. e. operations that achieve the same functionality. This approach proceeds in two steps. The first step consists in extracting operations from WSDLs files associated with web services. The second step consists in studying the similarity between the operations considering their outputs since we focus on the functional similarity and not the substitution (operations that replace some others). The focus on the outputs of an operation is the safest way to infer that two operations are functionally the same. Our approach is based on a semantic and syntactic analysis, and on a threshold parameter whose value has a direct impact on the quality of the results.

**REFERENCES**


[1] Douglas Martin, JamesR, Cordy., "Analyzing Web Service Similarity Using Contextual Clones," in *IWSC* 11, May 23, 2011.

[2] Dong, X., Halevy, A., Madhaven, J., Nemes, E. and Zhang, J., "Similiraty Seaurche for Web Services," in *Proceedings of the 30th VLDB conference*, Toronto, Canada, pp. 372-383, August 2004.

[3] Natalia Kokash, "A Comparaison of Web Service Interface Similarity Measures," *STAIRS*, pp. 220-231, 2006.

[4] T Rachad, J Boutahar, S.El ghazi. "A New Efficient Method for Calculating Similarity Between Web Services," *Journal of Advanced Computer Science and Applications*, vol. 5(8), pp. 60-67, 2014.

[5] Jaouad Boutahar, Taoufik Rachad, Souhail El houssaini "A New Efficient Matching Method for Web Services Substitutio," *Journal of Computer Science Issues*, vol.11(2), Sep 2014.

[6] Okba Tibermacine, Chouki Tibermacine, Foudil Cherif, "A Practical Approach to the Measurement of Similarity between WSDL-based Web Services," in *proceedings of the frensh-speaking conference on software Architecture (CAL'2014)*, France, 2014.

[7] Fulvio Corno, Teodoro Montanaro, Carmelo Migliore, and Pino Castrogiovanni, "SmartBike: an IoT Crowd Sensing Platform for MonitoringCity Air Pollution," *International Journal of Electrical and Computer Engineering (IJECE)*, vol.7(6), pp. 3602-3612, Dec 2017.

[8] Erick Fernando, Hetty Rohayani. AH, Pandapotan Siagian, Derist Touriano, "Analysis of Security and Performance Service in Service Oriented Architecture (SOA) and Data Integration," *Proceeding of International Conference on Electrical Engineering, Computer Science and Informatics (EECSI 2014)*, Yogyakarta, Indonesia, pp. 20-21, 2014.

[9] Teddy Surya Gunawan, Muhammad Hamdan Hasan Gani, Farah Diyana Abdul Rahman, Mira Kartiwi, "Development of Face Recognition on Raspberry Pi for Security Enhancement of Smart Home System," *Indonesian Journal of Electrical Engineering and Informatics (IJEEI)*, vol.5(4), pp. 317-325, Dec 2017.

[10] Jammel Mona, "Data Communication in Internet of Things: Vision, Challenges and Future Direction," *TELKOMNIKA (Telecommunication, Computing, Electronics and Control)*, vol.16(5), pp. 2057-2062, Oct 2018.

[11] Jesús Alvarez-Cedillo, Elizabeth Acosta-Gonzaga, Mario Aguilar-Fernández, Patricia Pérez-Romero, "Internet Prospective Study", *Bulletin of Electrical Engineering and Informatics*, Vol. 6(3), pp. 235-240, Sep 2017.

[12] MOHAMMED AHSAN, "Smart Walking Stick for the Visually Challenged," *Indonesian Journal of Electrical Engineering and Computer Science*, vol 12(3), Dec 2018.

[13] Hakim Adil Kadhim, Nabeel Salih Ali, Dheyaa M. Abdulsahib, "Management and Achieving System for Metal Detection Robot Using Wireless-Based Technology and Online Database Registry," *International Journal of Power Electronics and Drive Systems (IJPEDS)*, vol 10(1), Mar 2019.

[14] Pierluigi Plebniurbe, "BarbaraPernin: URBE: Web Service Retrieval Based on Similarity Evaluation," *IEEE Transactions on Knowledge and Data Engineering*, vol. 21, Nov. 2009 .

[15] Stroulia, E.and Y.Wang., "Structural and Semantic Matching for Assessing Web Service Similarity," *International journal of Cooperative Information System 14*, pp. 407-437. Explaining research.