A New Approach for Grouping Similar Operations Extracted from WSDLs Files using K-Means Algorithm

Rekkal Sara
LAPECI Laboratory
University of Oran 1 Ahmed Ben Bella;
Oran, Algeria

Amrane Fatima, Loukil Lakhdar
Computer Science Department, Faculty of Sciences,
University of Oran 1 Ahmed Ben Bella;
Oran, Algeria

***Abstract*—Grouping similar operations is an effective solution to the various problems, especially those related to research because the services will be classified by joint operations. Searching for a particular operation returns, as a result, all services with this same operation, but also the problems related to the substitution (such as, during a call failure or a malfunction). A list of similar operations is returned to the client. He chooses an operation, based on non-functional criteria. In this work, our goal is to study the functional similarity between operations, and thus constituting groups of similar operations, while benefiting from the K-means algorithm.**

Keywords—Web services; WSDL; inputs; outputs; similarity; syntax analysis; semantic analysis; Hungarian maximum matching; K-means

I. Introduction

The need generated by a client invoking a server application is at the origin of what is known today as a web service.

A web service is a solution to a given need, from a computer science point of view; a web service is an application that makes its features accessible via the Internet. It can be public or private.

Web services are based on SOA architecture (Service Oriented Architecture). The latter is based on three main actors: provider, directory and client (shown in Fig. 1).

The main advantage of this architecture is that the client does not need to know the service provider; he must simply express its need precisely in the form of a query querying the UDDI (Universal Description Discovery and Integration) directory.

Faced with this need, several web services may exist, so they are returned as a result: the customer then chooses the one that best suits his needs and he starts to invoke it.

Manipulation process seems simple, but as this technology is not yet mature, there are still many problems that require us to create effective solutions, such as: the search results must match the needs expressed, but this is not always the case, and this is due to:

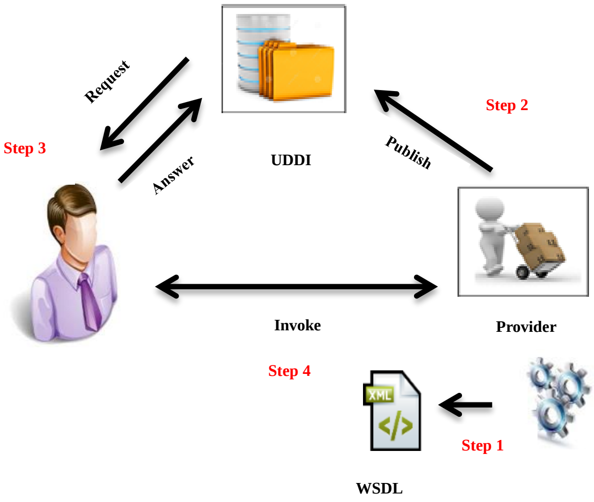

Fig. 1. SOA actors.

1) The continuous growth of the number of services deployed on the net complicates more and more the research task and also increases the search time.

2) The Web services are volatile; they often operate in a highly dynamic environment as that the providers can remove, modify, and relocate them frequently which causes a malfunction at the time of their use.

3) In general, the customer is not interested by all operations offered by the service, but by some of them.

So, in order to remedy this and to facilitate the discovery and the substitution, we propose to reorganize the web services space in a meaningful manner by constituting groups of (Services, Common Operations). The gains we get from this reorganization are:

1) An improved search time (quick and easy search): A simple correspondence with the operation and the result will be returned.

2) All results will be returned: All services corresponding to the need will be listed, and the client selects the one that suits him best according to non-functional criteria.

3) During a call failure of an operation or during a malfunction, a list of similar operations will be returned; the client selects the one that suits him the most, according to non-functional criteria.

In this document, we decided, as a first step, to focus on the study of similarity between operations of web services, and thus constituting groups of (Services, Common Operations).

The remainder of this document is administered as such: Section II, to introduce you to the related work, Section III, devoted to the presentation of the proposed approach, Section IV is dedicated to the presentation of the results of the experimentation, and finally, a conclusion is presented in Section V.

II. Related Work

Our goal, as mentioned above, is to build clusters of similar operations extracted from WSDLs files; and since this is the first initiative in the field, we have relied on works done on WSDLs files that respond to various problems related to the discovery, the composition, the substitution and the similarity between Web services using different methods and techniques.

Authors of [1] suggested a technique for lexical and structural similarity assessment of web services descriptions; their similarity study is based on the measurement of the similarity between descriptions (documentation) of various elements, but the majority of web services that we found are not documented, which means that the technique is not very practical.

The authors of [2] built a network so that the nodes represent the operations of the web services. A link joins two similar operations; the similarity is studied according to four functions. In the resulting network, similar operations are connected and form a graphical component. The authors summarize the similarity in only four cases and ignore many others significant cases, so the results risk of being not good enough.

In [3], web services are organized into substitutable service communities, as each community is associated to a specific functionality, so the web services meet the same need. This similarity has been defined through the similarity study of their operations. The authors in this work define mapping technique between services such as mapping between two Web services can be simple or complex. Simple mappings align one element (input/output parameter) to another. Complex mappings deal with incompatibilities of operation signatures, data types, data units, etc. Mappings (simple or complex) require the aggregation of some functions in order to convert units, currencies, and measurements as well as to perform data transformations, but these functions have not been identified or discussed.

Dong et al. in [4] 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. The authors consider only some elements such as parameters names, operations names and descriptions and ignore others elements, such as types.

In literature [5], authors have suggested an approach to determine the similarity between web services. To do so, 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 others syntactic ones. The authors use several metrics to calculate semantic and syntactic similarity where they had to choose just the best of them. Also, the authors did not calculate the precision of their method.

In [6], similar web services are clustered, the similarity study was based on the semantic comparison of elements extracted from WSDL files such as parameter names and operation names, using the Wu method. Authors consider some elements and ignore others, such as types.

III. Proposed Approach

A. Methods and Basic Tools

1) Syntactic and semantic analysis

In our work, the similarity between operations depends on the similarity between their descriptions extracted from WSDLs files, since it is difficult, even impossible, to access their source codes.

The description extracted from the WSDLs files is in a high-level language (human language). For this, we used semantic and syntactic methods considered better to evaluate them. These methods return similarity measures between [0, 1], such that 0 means dissimilarity and 1 means similarity. As long as we tend to 1 as long as the compared elements are more and more similar.

a) Syntactic analysis

Syntactic similarity is presented in this survey though different String-Based algorithms (Fig. 2).

String-based measures determine the similarity by operating on string sequences and character composition. The string-based methods are divided into: character-based and terms-based approaches. Algorithms of character-based similarity measurement consist of Smith-Waterman, N-gram. Damerau-Levenshtein, Jaro–Winkler, Needleman–Wunsch, Jaro, and Longest Common Substring (LCS). Algorithms of term-based similarity measurement include Block Distance, Cosine similarity, Dice's coefficient, Euclidean distance, Jaccard similarity, Matching Coefficient and Overlap coefficient [7].

The works led by [8], [9] concludes that Jaro-Winkler has performed better in term of results in several experiments and can be used in several fields and it is much faster than the others methods.

In this work, we choose to use the Jaro-Winkler method.

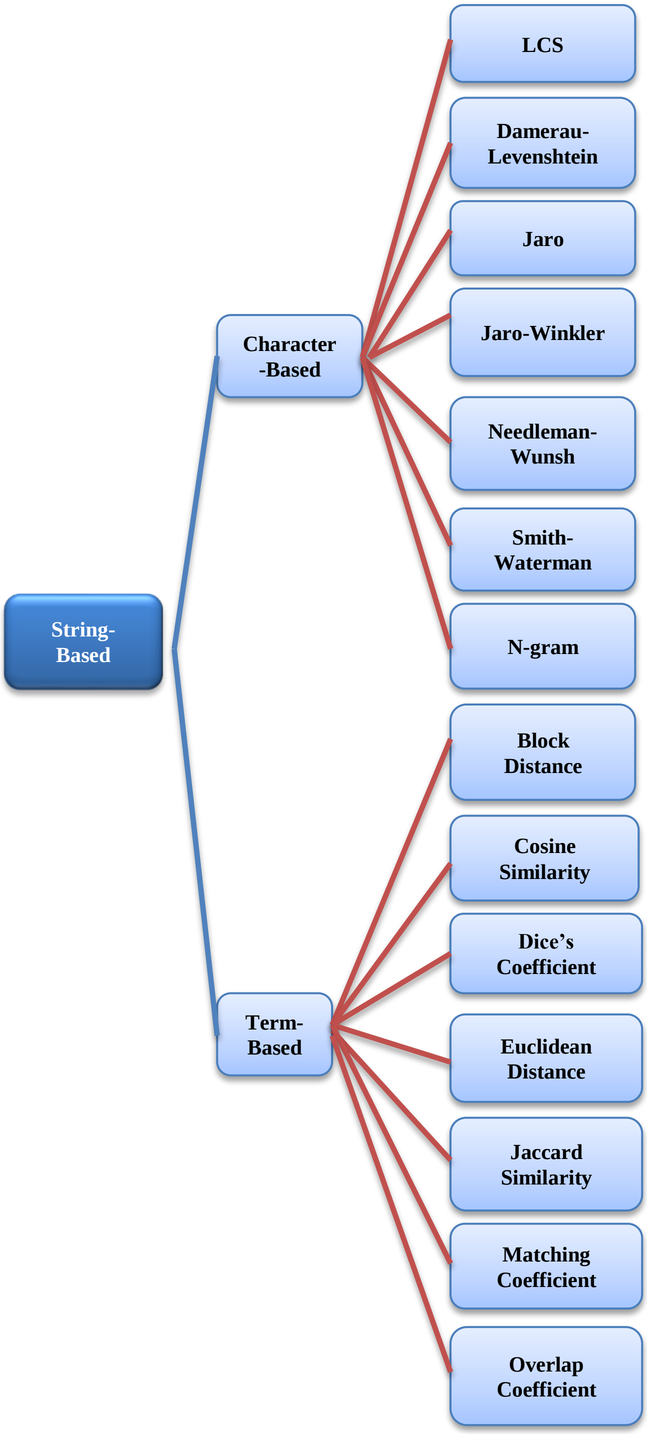

Fig. 2. String-based similarity measures. [7].

b) Semantic analysis

These methods consist of assigning to a pair of words, a metric based on the similarity of their meaning. According to [10], these methods can be classified into three categories:

1) Evaluating similarity by counting edges

Consists in calculating the distance between two concepts in a taxonomy as WordNet, by the shortest path, in other words, it evaluates the number of semantic links separating the two concepts in the ontology. There are several methods, the most known are: Rada (1989), Lee (1993), Wu & Palmer (1994), etc.

2) Evaluating similarity by Information Content

The informative content of a concept reflects the relevance of a concept in the corpus, taking into account the frequency of the appearance of the words to which he refers, as well as the frequency of appearance of the concepts he generalizes. There are many methods, the most well-known are: Lin (1998), Resnik (1995), Jiac (1997) etc.

3) Hybrid approach

It is a combination of the similarity measures mentioned above. Parameters Length, depth, and local density form a part of the nonlinear function which measures the similarity between concepts [11]. Among these methods are: Jiang and Conrath (1997), Lec (1998).

According to [8], it cannot be said that there are more efficient or more optimal methods than others because the studies conducted to examine them took some evaluation criteria and ignored others. But the method that provides better results with WordNet is Wu - Palmer. Thus, it has the advantage of being simple to implement and also have good performance, compared to other similarity measures.

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 maximums weights in a bipartite graph. Mathematically the problem can be formulated as follows:

Let G (X, U) be a bipartite graph, Fig. 3, of which:

- X = (P ∪ Q) set of nodes of the graph.
- U = set of links connecting the nodes characterized by costs.

= [f (q1, p1) +f (q2, p3) +f (q3, p2)]/3

= [1.00+0.7+1.0]/3=0.9

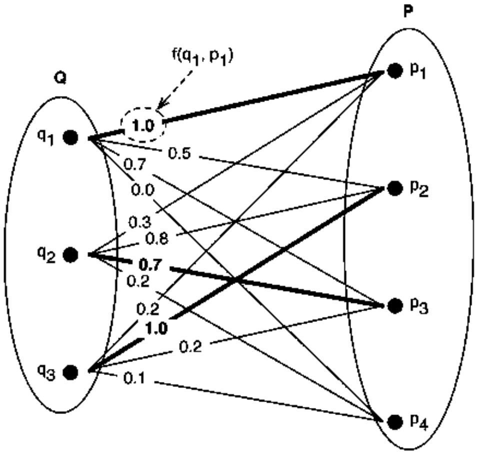

Fig. 3. Bipartite graph problem [12].

TABLE I. MATRIX MODELING THE BIPARTITE GRAPH ABOVE

1.0	0.5	0.7	0.0
0.3	0.8	0.7	0.2
0.2	1.0	0.2	0.1

A graph can be represented by a matrix (Table I), 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) The K-means algorithm

The k-means algorithm is a method of clustering; it allows us to group similar objects. It aims to divide n individuals into K subgroups, as homogeneous as possible; K is a fixed number by the user.

The procedure follows five steps:

- Choose a number K forming K clusters.
- Choose centers Mi (i=1…n) for each cluster (as much as possible far away from each other).
- Assign each object O to the cluster Ci of center Mi such that distance (O, Mi) is minimal.
- Recalculate Mi of each cluster (the center of gravity).
- Return to the third step if you have made an assignment.

The algorithm stops when:

- Two successive iterations lead to the same partition.
- We define stopping criteria such as the maximum number of iterations.

a) Advantages

- The k-means algorithm is very popular because it is very easy to understand and to implement.
- Its conceptual simplicity and speed.
- Applicable to large data sizes, and also to any type of data (even textual), just by choosing a good notion of distance.

b) Disadvantages

- The number of classes must be defined at the beginning.
- The result depends on the initial draw of the class centers.

B. WSDL file

As its name indicates, WSDL is used to describe web services. It is divided into three major elements that can be separated and used independently or combined to form a unique XML (eXtensible Markup Language) document. These elements are:

- The elements: Types, Message, PortType, and Operation: define the operations offered by a web service and the inputs and outputs of each of these operations.
- Binding element: Defines the communication protocols and Internet transport used to invoke the operations defined in the PortType element.
- The elements: Service and Port: Defines access points to the service.

As mentioned earlier, a client looks for a service for the operations he carries out. More precisely, a client is only interested by the results produced by the operations. So we are interested, in this study, at the extraction, from each operation the following elements:

1) Operation’s name.
2) Output message name.
3) The outputs parameters and their associated types.

Our choice of parameters is justified by:

- In general, a customer is looking for an operation that produces outputs that he needs.
- We are interested in studying the functional similarity of operations. Two operations that produce the same result necessarily mean that they do the same work (the same functionality) and they meet the same need.
- The authors of [13] have developed a theory of substitutability, such that two Web services are substitutable: if one requires as many or fewer inputs and produces as many or more outputs, which means that the focus is more on the outputs than the inputs.

C. Similarity Process between Operations

Similarity process between operations consists to:

- Extract necessary elements from WSDLs files.
- Transform complex parameters into simple parameters.
- Evaluate the similarity between operations.
- Construct groups of similar operations using the K-means algorithm.

1) Extract necessary elements from WSDL files

Extract from WSDLs files the following elements (see Fig. 4):

- Operation identifier.
- Output message identifier.
- Output parameters identifiers, and their associated types.

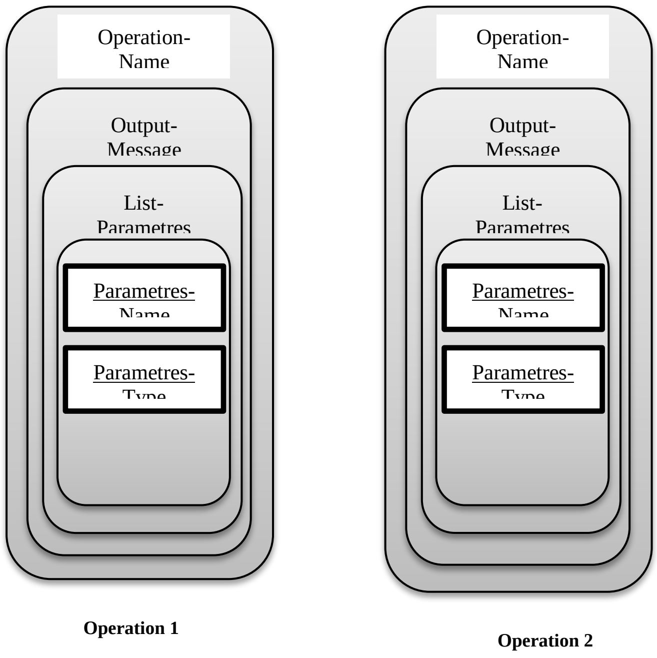

Fig. 4. Similarity between operations.

2) Transform complex parameters into simple parameters

As mentioned earlier, the WSDL file includes the description of several elements, whose output parameters, on which our similarity study is based.

These parameters can be of simple type (Identifier + Type) or complex (Parent identifier + identifiers of sub-elements + Type).

Comparison of simple parameters doesn't pose a problem, unlike complexes. To remedy this, we have to transform the complex parameters into simple parameters, by aggregating the identifiers of the sub-elements with the parent identifier (see Fig. 5 and 6).

3) Evaluate the similarity between operations

Let O1 and O2 be two operations extracted from different services S1 and S2.

a) Similarity process

i) Similarity calculation

In our work, the similarity between operations is measured by the following function:

Sim = [Sim_Msg () + Sim_Ops_Name ()]/2.

Where:

- Sim (): is the main function, which calculates the similarity between two operations.
- Sim_Ops_Name (): is the function that measures the similarity between the identifiers of the compared operations (see Section III-C-3(*a*)(*ii*)).
- Sim_Msg (): is the function which measures the similarity between two outputs messages such that:

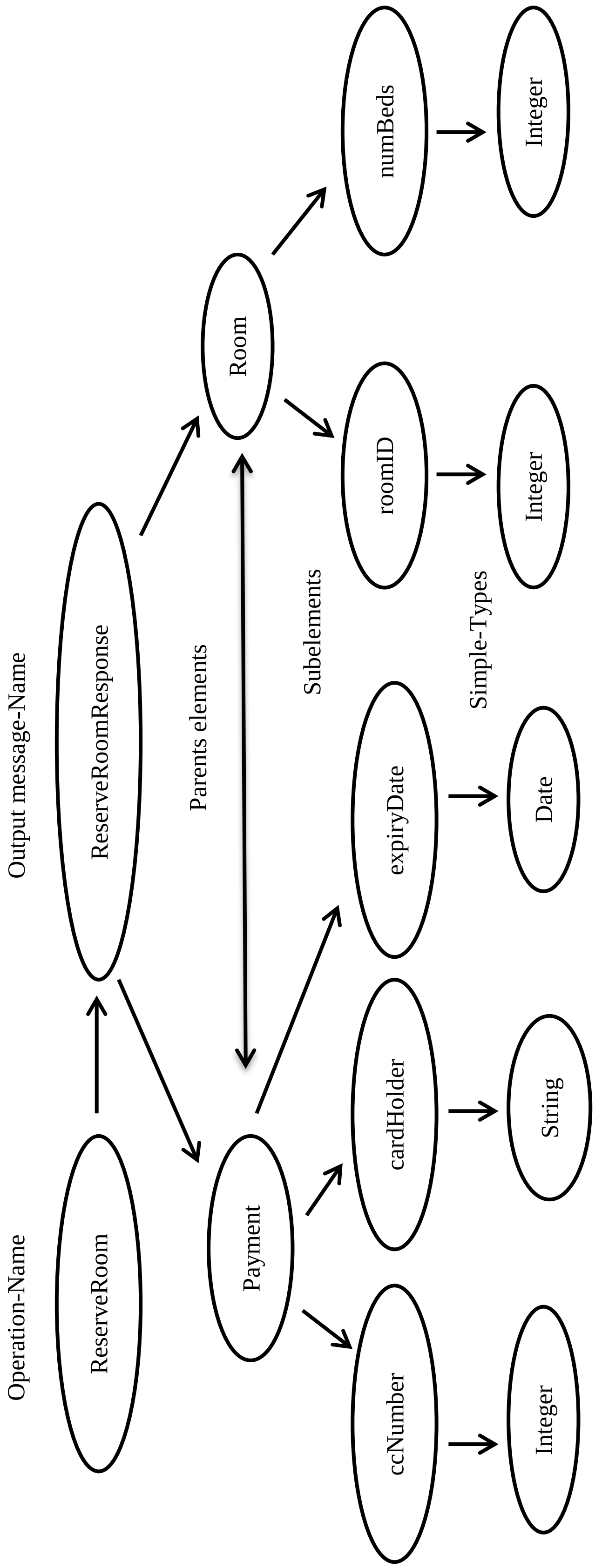

Fig. 5. The description of the ReserveRoom operation from hotel reservation service before aggregation.

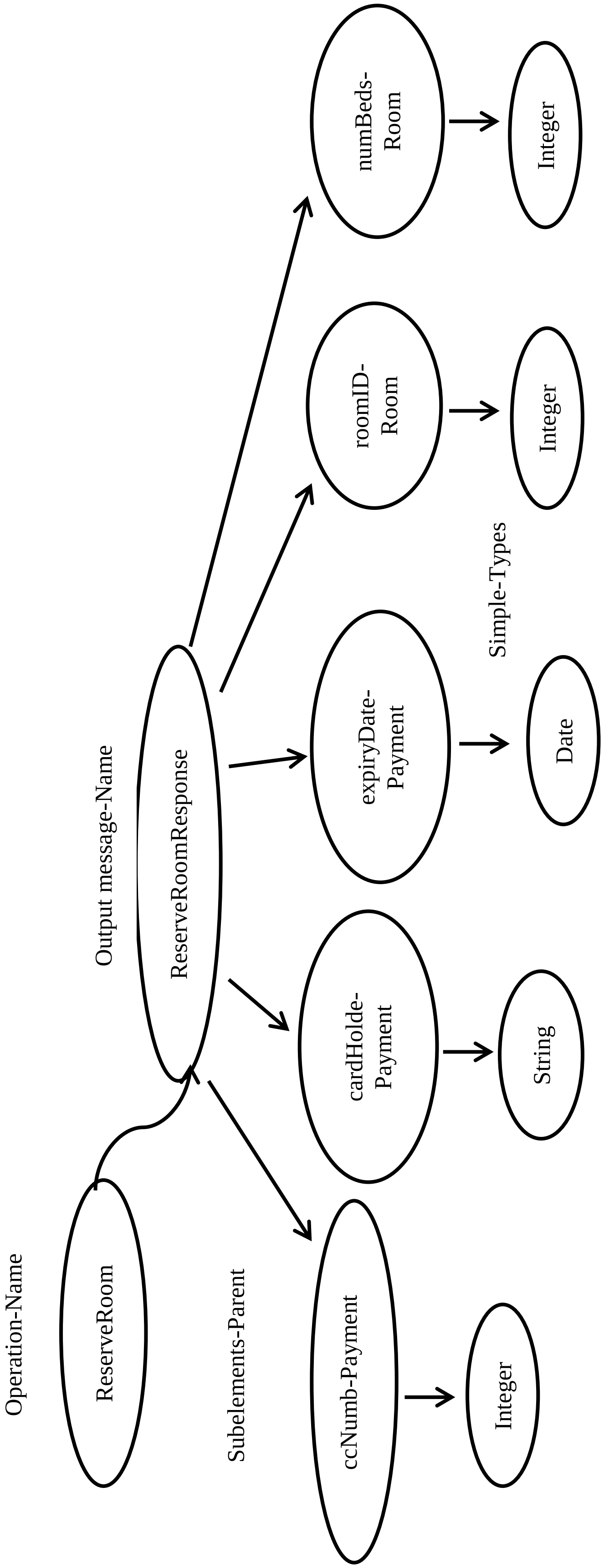

Fig. 6. The description of the ReserveRoom operation from hotel reservation service after aggregation.

Sim_Msg = [Sim_List_Pars () +Sim_Msg_Name ()]/2.With:

- Sim_Msg_Name (): is the function that measures the similarity between the identifiers of the compared outputs messages (see Section III-C-3(*a*)(*ii*)).
- Sim_List_Pars (): is the function that measures the degree of similarity between parameters. This function consists of:

A. Building a similarity matrix whose lines refers to the parameters of the first operation and the columns refer to the parameters of the second operation (Table II). The following formula determines the values of this matrix:

Sim_Pars= [Sim_ident () +Sim_Types ()]/2.

Where:

- Sim_Pars (): is the function that calculates the similarity between parameters. This measure is included between [0-1], where 0 means the dissimilarity of the compared parameters and 1 means their similarity. As long as we tend to 1 as long as they become more and more similar.

Where:

- Sim_ident (): is the function that measures the similarity between the identifiers of the outputs parameters (see Section III-C-3(*a*)(*ii*)).
- Sim_Types (): is the function that measures the similarity between the types of parameters (see Section III-C-3(*a*)(*iii*)).

B. Calculate the degree of similarity between the list of parameters, by calculating the average of maximum scores (the Hungarian method).

TABLE II. SIMILARITY BETWEEN PARAMETERS

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

ii) Similarity between identifiers

An identifier (parameter name or message name or operation name) is a word or a sequence of concatenated words.

Measuring the similarity between two identifiers consists of:

- Chopping identifiers into words, (in the case of an identifier composed of several words).
- Remove stop words, as well as special characters and numbers.
- Extending the abbreviations.

- Lemmatizing the segments (use the singular, the infinitive for verbs, the masculine for adjectives, etc.).
- 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 III. 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).
- Calculate the degree of similarity between the identifiers, by calculating the average of maximum scores (the Hungarian method).

TABLE III. SIMILARITY BETWEEN TWO IDENTIFIERS

Identifier1 Identifier2	Word1	Word2	Word 3
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

iii) The similarity between types

The type T of an identifier is: Integer, Real, String, Date or Boolean. In [12] and [14], authors propose Table IV, which determines the similarity between the different possible types:

TABLE IV. SIMILARITY BETWEEN TYPES [12]

	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 with the following formula:

Sim_Types = min [Sim (T1, T2), Sim (T2, T1)].

Where,

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

4) Constitute groups of similar operations using the K-means algorithm

a) Concept of distance between operations

The distance between two objects is defined by their convergence or divergence from each other.

In this work, the distance between two operations is defined by their degree of similarity. this degree is between [0, 1] where 0 means dissimilarity and 1 means similarity of operations, as long as we tend to 1, as long as they become more and more similar. For this, a threshold was defined from which we consider that the compared operations are very similar (very close to each other). This has been defined by experts, who concluded that from a threshold of 0.7, the operations can be considered very similar.

b) Cluster's Center

The K-means algorithm consists of determining for each cluster Ci, a center Mi, so an object O (operation) will be assigned to the cluster whose distance (Mi, O) is minimal. In our case, the cluster's center is represented by an operation called the representative operation, by default, is the first operation assigned to the cluster Ci, so an operation Oj will be assigned to the cluster Ci if, and only if, the distance (Mi, Oj)> = Threshold.

c) Application of K -Means algorithm

K-means algorithm mentioned previously group similar objects into a single group. The major disadvantage of this algorithm is the number of classes K which must be fixed at the beginning, also the random selection of the objects forming centers'. So to remedy this, the following solution has been proposed:

```
Choose a number K forming K clusters.
  For i =1 to n (Total number of operations) do
  Choose a random operation. Let Oi.
     While: Oi is not assigned yet, do:
          Study the distance between Oi and the
          representative operations of the clusters (at the
          beginning, all the clusters are empty).
              If it is similar, assign it to its cluster, return
                to 1.
              Else:
                  If there are empty clusters, assigned to
                  one of these clusters, it becomes its representative element, return to 1.
                  Else (if there is no empty cluster):
                      Increase the k, create a new
                      cluster.
                      Assign this operation to this new
                      cluster; it becomes its representative
                      element, return to 1.
     End while.
  End for
   Count the number of empty clusters, delete them and decrement the K.
```

d) Stop criteria

All operations are assigned to their clusters.

e) Temporal complexity

The complexity of the worst case is n * m. such as:

- n: represents the total number of operations.
- m: the number of non-empty clusters

IV. EXPERIMENT RESULTS

A. The WSDLs Files Used

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

B. The WSDLs Files Used

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

C. Results and Assessment

The tool has been experimented with 15, 30, 49 WSDLs samples successively with a number of operation, 36, 150, 300 and a number of parameters in the same order 137, 627,860.

To evaluate our results (the formed groups), we compared our results with those obtained of experts experience on the same sample and we calculate the precision and recall. The result is shown in Table V.

TABLE V. RECALL AND PRECISION MEASUREMENT

	15 WSDLs	**15 WSDLs**	**15 WSDLs**
Precision	1.0	1.0	0.98
Recall	1.0	0.97	0.94

D. Discussion of Results

Table V shows the results obtained which are described by two measures: precision and recall.

1) Precision: This 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.

2) Recall: This measures the proportion of all the correct results that a software might theoretically find, and is the ratio of the number of relevant elements found by the total number of relevant elements.

According to Table V, the results obtained (precision & recall) are very satisfying, which means that the groups formed by our software which implement our method, are very close to those obtained of experts experience on the same sample, this indicates that our method is very close to the human evaluation, and confirms the effectiveness and the reliability of our approach.

V. CONCLUSION AND FUTURE WORK

In this work, our goal has been, as a first step, to focus on the study of similarity between operations of web services, and thus constituting groups of (Services, Common Operations).

In this work, we rely on the outputs, as two operations are considered functionally similar if, and only if, they produce the same outputs (same results). We used semantic and syntactic methods considered as better; also we arranged the K-means method to group the similar operations.

Our future work consists of proposing a research approach adapted to this new space while proving that this reorganization of the Web services space as discussed previously in this article, improves search time and facilitate the discovery compared to the different existing approaches.

REFERENCES

[1] Natalia Kokash, “A Comparaison of Web Service Interface Similarity Measures”, STAIRS, p.220-231, 2006.

[2] Chantal Cherifi,Vincent Labatut, Jean-François Santucci “Topological Properties of Web Services Similarity Netweorks”, Journal of Strategic Advantage of Computing Information Systems in Enterprise Management, ATINER, pp. 105-117,2010.

[3] Y. Taher, D. Benslimane, M.-C. Fauvet, and Z.Maamar,” Towards an Approach for Web Services Substitution”, in 10 th International Database Engineering and Applications Symposium,2006,pp.166-173.

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

[5] 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

[6] Konduri, A.&C. Chan. “Clustering of web Services based on WordNet semantic similarity”, 2008.

[7] Wael H. Gomaa, Aly A. Fahmy, “A Survey of Text Similarity Approaches,” International Journal of Computer Applications, pp. 13-18, 2013

[8] 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, no 08, p. 60-67, 2014.

[9] Cohen, W., Ravikumar, P. and Fienberg, S., “A comparison of string metrics for matching names and records”, Proceedings of the International Conference on Information Integration on the Web (2003) Pages 73-78.

[10] Sonia Lajmi, “Annotation et recherche contextuelle des docuents multimédias, Allemagne : éditions universitaires europeennes 2012, 276 p.

[11] Atul Gupta, Dharamveer kr. Yadav “Semantic similarity measure using information content approach with depth for similarity calculation”, International journal of scientific & technology research, Volume 3, February 2014.

[12] Pierluigi Plebniurbe, BarbaraPernin: URBE: Web Service Retrieval Based on Similarity Evaluation” IEEE Transactions on Knowledge and Data Engineering, Vol. 21, Nov. 2009.

[13] Kona, S., A.Bansal, L.Simon, A.Mallya,G. Gupta and T.Hite. “USDL: A Service-Semantics Dscription Language for Automatic Servic Discovery and Composition”, 2006.

[14] Stroulia, E.and Y.Wang: “Structural and Semantic matching for assessing web service similarity”, International journal of Cooperative Information System14, 407-437.