Understandability Metric for Web Services

Usama Maabed, Ahmed El-Fatatry, Adel El-Zoghabi
Institute of Graduate Studies and Research, Alexandria University
Alexandria, Egypt.
e-mail: umaabed@alexu.edu.eg, e-mail: elfatatry@alexu.edu.eg, e-mail: zoghabi@alexu.edu.eg

Abstract-- This work is concerned with developing a quality index for Web service understandability. A major reason for poor search results in Web service repositories is the lack of proper descriptions for existing services and as a result such services are not counted in search results. We present a mechanism to specify and measure Web service interface quality. The mechanism includes metrics for both the registration and operation phases. The evaluation results show significant enhancement in the discovery process as a result of using the proposed mechanism.

Keywords—SOA; Web services; Usability; Discovery; Web service interface quality.

I. INTRODUCTION

Web service discovery is a major challenge for automatic Web service selection and integration. Discovery depends not only on Web service availability but also on the ability to understand the objective and the function of the available Web services, using the descriptive details. Web service description is often overlooked when working with a Web service to increase its quality. Even if a Web service follows all recommendations it will still be inaccessible if its description is difficult to comprehend. Several approaches and techniques have been proposed to address this challenge, including syntactic-based and semantic-based approaches [1][2]. Such approaches assume that correct and valid service descriptions already exist. In practice, the reality is different.

We analyzed more than 35,000 Web services from eight well-known datasets [3]-[7]. In these datasets, we found that only 49% of registered services are active. Further analysis of Web Services Description Language (WSDL) based Web services shows that only 19 % have a Web service description. Therefore, 81 % of Web services are not being considered in the syntactic discovery phase. Our findings are also supported by previous studies [8][9].

On other hand, the semantic approach suffers from “a cold-start problem” because it assumes that a corpus of previously annotated services is available [1] [2] [10].

Recently, Quality of Service (QoS) in Web service discovery has gained considerable attention as a vital research topic [8][11][12]. In this context, it is important to distinguish between two research domains within Web service discovery. The first approach focuses largely on Web service functionality and performance, such as response time, latency, availability, accessibility, and security. The second approach focuses on the quality of a Web service interface, largely assessing the Web service interface in terms of complexity. The importance of the second approach lies in the fact that without easily understandable Web service interface functionality, it is not reasonable to expect successful discovery or usability [11]. In general, the more details we can obtain about a Web service function and its domain, the more it becomes reachable and usable.

Analyzing a text or web site for its readability has a long tradition in literature. Different approaches and metrics have been employed to assess the readability of web sites, or to filter documents that match a user’s reading ability [13]. However, there are some differences between a web site and Web service description. On the one hand, text in the web is presented is usually long and presented differently than a Web service description, which expected to be short and straightforward. On the other hand, Web service description needs to be comprehended not only by humans but also by agents in case of auto-discovery.

Given the differences in the presentation and nature of Web service, having an index that gives an indicator on the readability or clarity of Web service interface, becomes more essential. In this regard, we have also to consider the majority of currently deployed Web services that have no description or that have a poorly written description. For such cases, focusing only on the readability of the Web service description might not help in identifying the Web service domain or its functionality.

Our research aims at introducing a new approach for fixing the current problem of poorly written Web service descriptions and providing practical control over Web service interface quality to minimize and avoid such bad practices.

In this paper, we address this topic under several aspects. First, we consider the problem of text noise in Web service description and the attribute names used in the Web service interface. This noise represents additional text that are typically not part of the main context and might not help in classifying or understanding the Web service function. Obviously, this noise should not be considered when determining the readability or clarity of a Web service interface. One way to eliminate the noise is to provide hand designed filters for cleaning the interface contents of a particular Web service. The second step is to have a mathematic approach to measure the understandability of the Web service interface. In this regard, we considered the number of the extracted meaningful words from the filtered Web service interface. We used the well-known WordNet [14], which is an electronic lexical database that is available to researchers in computational linguistics, text analysis, and many related areas. We analyzed the web-service operations and input/output message parameter meaningful naming as part of the Web service usability and understandability characteristics. In addition, we considered sharing the clarity and understandability indexes to be available during the discovery phase as part of QoS improvement.

This approach will help service providers in improving their service quality by satisfying discovery needs and measures. Service registry moderators and brokers will gain the advantage of providing valid and well-defined Web services with better quality, based on better classification and clustering that supports service discovery and composition. Consumers will effectively use the shared interface quality metrics during registry queries to select services that match their quality needs and development constraints with minimal effort.

Thus, the main contribution of this work is to develop a novel approach that addresses currently poorly written Web service descriptions with a focus on developing the required control that can help in addressing such malpractice.

The rest of this paper is structured as follows: the related work and previous developed metrics are discussed in Section 2, followed by the research problem in Section 3. The solution requirements are defined in Section 4. In Section 5, we introduce the proposed framework. The experimental Evaluation and the research findings are discussed in Section 6. Finally, the conclusion and recommendations for future research are presented in Section 7.

II. Related Work

In software quality research, metrics for service interface quality have only recently gained attention [15]-[18]. In this Section, we summarize efforts in the area of measuring the quality and complexity of a Web service interface.

The first work on Extensible Markup Language (XML) metrics was presented in [19]; the work presents a set of five metrics based on Document Type Definition (DTD) specifications and the schema graph representation, adopting common metrics used for software to determine the characteristics of DTDs. The authors use product metrics in terms of size, structure complexity, structure depth, Fan-In, and Fan-Out.

McDowell et al. [15] focus on XML schema types and introduce two indices for measuring quality and complexity based on eleven metrics. The proposed metrics are Number of Complex Type Declarations, Number of Simple Type Declarations, Number of Annotations, Number of Derived Complex Types, Average number of Attributes per Complex Type Declaration, Number of Global Type Declarations, Number of Global Type References, Number of Unbounded Elements, Average Bounded Element Multiplicity Size, Average Number of Restrictions per Simple Type Declaration, and Element Fanning.

Lammel et al. [20] proposed a suite of metrics for XML Schema that primarily focused on schema size and complexity. The suite ranges from simple counters of various types of schema nodes, such as Number of global element declarations, Number of global complex-type definitions, Number of global simple-type definitions, and Number of global attribute declarations, to more involved metrics such as McCabe, depth, and breadth. Their work helped to introduce the fundamental metrics for the XSD language and identified the basic feature model of the XSD language at a basic level [21].

Qureshi et al. [16] focused on measuring the complexity of XML documents based on different structural characteristics. Their work aims to lower the complexity of XML documents and improve their reusability and maintainability. They used the Weight Allocation Algorithm (WA) and the Document Object Model (DOM) for tree representation. Weights are assigned to each element of XML trees, according to their distance from the root node. The algorithm provides a mechanism to gauge the quality and comprehensibility of XML documents.

Visser [22] presents a more advanced metric that considers the structure of a schema by adopting well-known measurement methods from graphs. The metrics are adaptations of existing metrics for other software artifacts, such as programs and grammars. As a prerequisite, a graph representation must be computed from a given XML Schema to measure, for example, how closely the graph structure is related to a tree structure in which the measures of reclusiveness are identified.

Basci and Misra [18] developed a structure-based metric that measures the complexity related to the internal architecture and recursion of XSD components. The metric has been empirically and theoretically validated using 65 public Web services.

H. Sneed [23] proposed a suite of metrics that contains various metrics to assess the complexity of service interfaces and to determine their size for estimating testing effort. The proposed metrics range from common size measurements such as lines of code and number of statements to metrics for measuring the complexity and quality of Web Services. The relevant metrics are Interface Data Complexity, Interface Relation Complexity, Interface Format Complexity, Interface Structure Complexity, and quality-related metrics covering Modularity, Adaptability, Reusability, Testability, Portability, and Conformity. Sneed defines the complexity of a service interface as the median value of its lingual complexity and its structural complexity. The lingual complexity of a service interface has been defined in terms of type and number of occurrences, and the structural complexity defined in terms of entities and relationships, in which entities are the instances of the data-types, messages, operations, parameters, bindings and ports defined in the schema. A relationship refers to compositions and cross-references.

Kumari and Upadhyaya [24] proposed an interaction complexity metric for black-box components based on measuring the complexity of their interfaces. The parameters they considered are the size of each component and the interaction with the component in terms of input/output interactions. Graph theoretical notions have been used to illustrate the interaction among software components and to compute complexity. The proposed measure has been applied to five cases chosen for their study and yielded encouraging results, which may further help in controlling the complexity of component-based systems to minimize both integration and maintenance efforts.

The review of the literature reveals that few studies focus on developing metrics that measure the quality of Web service interfaces. Existing metrics partially address this subject by focusing on the WS interface structure; these depend on counting a schema's components, or measuring the schema's complexity by considering different weights for each schema component. Although these metrics are important, they do not yield sufficient information about the clarity of a given interface and the understandability of each independent component. For instance, the count of service elements and types is more relevant to the service context and its functionality, which might vary from one domain to another. In addition, measuring the complexity of a schema's complex types or elements recursively by assuming that each of its sub-components has its own weight requires extra effort with no justified effect on the overall complexity measurement. To conclude, the metrics that measure schema complexity by counting components do not yield sufficient information about the complexity value of a given schema [18].

III. Problem Definition

Research into Web services has traditionally concentrated on issues such as service semantics, discovery, and composition. However, there has been little focus on currently deployed Web services. Investment in such services is far from efficient.

Although many efforts have been invested in finding new approaches to improve the semantic description of WS, most service descriptions that exist to date are syntactic in nature [25].

In an SOA context, Web service interface-related quality problems include poorly written Web service descriptions and a shortage of quality metrics for Web service interfaces. These two major issues also directly affect usability and the discovery process. This has been supported by different research findings. Zheng et al. [26] concluded that the WSDL files on the Internet are fragile. WSDL files may contain empty content, invalid formats, invalid syntax, and other types of errors.

In this Section, we summarize the defined issues with current Web services implementations, which is supported by previous researches.

A. *Web Service Description Poor Quality*

A major limitation of the Web services technology is that finding and composing services still requires manual effort. Although semantic web technology appears to be a promising approach for automated service discovery, it has several limitations [25] that can be summarized as:

- Most existing service descriptions are syntactic in nature.
- The vast majority of already existing Web services are specified using WSDL and do not have associated semantics.
- It is impractical to expect all new services to have semantic descriptions.
- From a service requester's perspective, the requester may not be aware of all terms related to the service request or domain knowledge.
- Introducing semantics into Web services has not yet moved to the industrial implementation phase.

Conversely, a search for semantic service descriptions conducted by Klusch and Xing [27] with a specialized metasearch engine, Sousuo, found no more than approximately 100 semantic service descriptions in prominent formats such as OWL-S, WSML, WSDL-S, and SAWSDL on the web. This number is very small compared with more than half a million sources indexed by the semantic web search engine Swoogle, and several hundreds of validated Web service descriptions in WSDL found by Sousuo on the web [25].

Having a quality model with suitable metrics has become crucial for assessing the Web service interface quality and its readiness for public use.

IV. Solution Requirements

In our research, it was necessary to measure a number of issues that have not been addressed or fulfilled, or have not been identified in Web services discovery. The main issues are currently registered but invalid Web services, poorly written WS descriptions, low usability, and the issue of sharing WS interface quality metrics. Controlling WS interface clarity and understandability is an important objective because it affects all other quality attributes such as usability, reliability, and maintainability. The following points represent our proposed approach to achieve better usability with a focus on WSs with poorly written descriptions.

A Web service interface specifies all of the information needed to access and use a service. The description should be rich, containing sufficient details and aspects of the service. Efficient and successful implementation of WSDL requires quality control with relevant metrics to ensure the required quality. The implementation details of a Web service are hidden behind their interfaces, which are published on the Internet and can be accessed through WSDL. WSDL consists of a set of operations, which are the access points for interacting with the outside computing environment. Such interfaces require metrics that are more relevant to its nature as a black box, with no access to its implementation source code.

To achieve a Web service with a high level of usability, we must consider the following.

- Describe the Web service functionality properly.
- Define Types and elements globally, facilitating reuse in other XML schemas and in the same XML schema documents.
- Design the Web service interface structure in terms of traceability and understandability.
- Use proper naming for attributes and elements within the schema documents and annotations.

V. Proposed Approach

Our approach to enhance Web service discovery and usability has two facets. One empowers the current WS descriptions using the extracted words from the types and element names. The second is the use of ISO/IEC 25010 standard [28] to develop a consistent model to evaluate Web service interface quality. We extend recent research efforts that focus on developing WS interface complexity metrics, and introduce clarity and understandability metrics. This approach addresses a Web service interface with a poor description by validating its clarity and quality, extracting distinct tokens from the used attributes and element names, and sharing this information through WSDL schema extension. The information is then utilized by service providers to improve their service quality and by consumers during discovery and selection.

A. Dataset used in this research

Throughout this research, we conducted several large-scale evaluations using popular WSDL datasets, which were obtained from well-known Web services providers. Because the main objective here is to improve the WS usability by consider its reachability through different discovery approaches, we consider some non-WSDL-based approaches during our analysis with initial focus on the syntactic part.

The QWS dataset [3] represents 2507 real WSDL files obtained from public sources on the Web. OPOSSum [6] is a database of service descriptions that has 1263 WSDL files. WS-DREAM [7] is a dataset that is composed of 3738 WSDL files that aim to reflect real-world data for Web Service researchers. Service-repository [4] is a directory of SOAP Web services. Servicefinder, Biocatalogue, Seekda, and Xmethods [29] are Web services catalogues.

We analyzed these datasets not only to support our research but also to provide a reusable dataset and practical feedback for promoting research on Web services interface quality. We applied the following procedure during our research:

- First, we tested the Web services availability and isolated the non-active Web services. The aim was to identify the bad files that had no active services, files referencing unavailable schema, files with local file references to schema files, or incomplete files.
- Second, we analyzed the active Web services to determine the problem size of Web services with no description. In this phase, we checked whether the published Web services had a description part. We also assessed the number of the meaningful words it contained. Meaningful words refers to words that cannot be considered part of stop-words, or terms which commonly appear across many service descriptions and can be found in a lexical database. The meaningful number of words is used to identify the Web service domain or classification or for clustering analysis. During this process, we addressed the duplication of Web service definitions that exist in some datasets. Primarily, we utilized the service name, host, and the Web service structure in identifying the duplications.

B. Web Services availability and descriptions

In this Section, we focus on two criteria: WS availability and WS description. We analyzed these parameters using previously defined datasets to identify active Web services that are valid for our research. Our analysis results are depicted in Fig. 1, which shows that the average percentage of active WSs is approximately 49%.

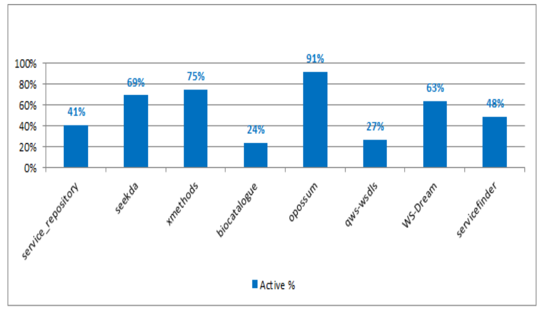

Figure 1. Active WS Percentage per Dataset

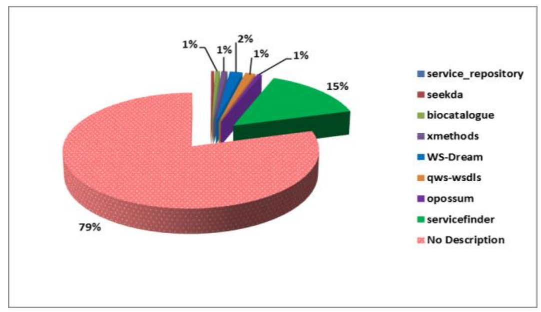

Figure 2. WSs with Description vs. WSs without Description

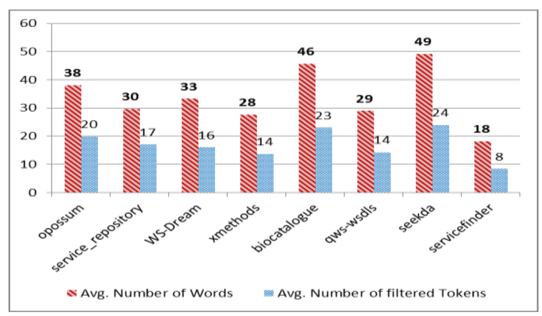

Figure 3. WS description - Avg. words vs. Avg. Token

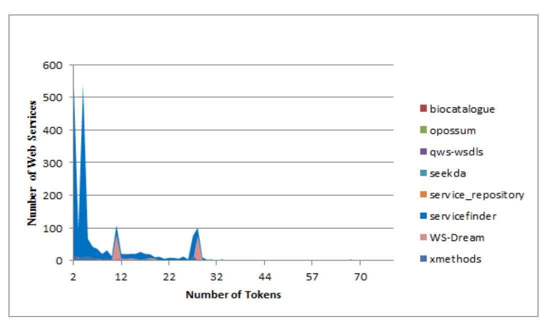

Figure 4. Number of Tokens Distribution

To understand the missing description problem of Web services, we considered in our analysis the complete collected active Web services after removing the Web services duplications. Fig. 2 shows that the total number of Web services that have descriptions is only 21% of the overall collected Web services.

Having collected the number of WSDL files with descriptions, the second step was to inspect the content of each file to evaluate the description part. To inspect the description, we applied the Information retrieval technique for removing the 'stopwords'. We used the stopwords to eliminate the terms that commonly appear across many service descriptions. For Web services with descriptions, Fig. 3 shows a comparison between the number of words found in the description part and the number of the extracted meaningful tokens.

As shown in Fig. 4, we found that the number of tokens in the Web service description ranges from 2 to 33 tokens. This shows that having a Web service description as a single criterion is insufficient to meet the quality requirement; there is also a need for a control on the size and meaning of the parts to ensure clarity and understandability requirements are met. This control must maintain both the minimum number of meaningful words contained in the Web service description and the operation names.

C. Web Service Interface Quality Metrics

Web service interface quality refers to the effort required to understand its functionality, structures, and the messages that are responsible for exchanges and for conveying Web service data between the service requestor and the service provider. Web service interface complexity can be measured by analyzing the XML Schema structure embedded in the WS interface. The focus of this Section is to propose a set of WSDL schema metrics to fill the gap caused by the missing clarity and understandability part of the WS interface quality control.

1) WS Interface Clarity and Understandability

In the previous sections, statistics show that only 21% of Web services have descriptions. In other words, approximately 80% of Web services are not considered during the normal syntactic search and discovery process. In this Section, we explain a new approach that depends on the parameter names of Web services: Web service name, endpoint, messages, and schema types. Throughout this research, we use the term "Clarity" to refer to the extent to which a Web service and its operations' naming are developed employing an appropriate and clear standard that is easily comprehended during the discovery process. This influences the understandability of the WS interface.

The Clarity Index measures the degree of syntactic understandability by parsing the Web service interface and extracting the meaningful parts (tokens) from the names of services, operations, and schema types. To calculate the Clarity Index, we compare the number of the extracted meaningful tokens to the overall number of terms used in these names for the overall elements used in the Web service interface. The same is applied on the service description part, which is measured by the following formula.

$$\mathrm{Clarity}_{\mathrm{Index}} = \left(\frac{\sum_{i=0}^{n} t_s}{\mathbf{n_s}} + \frac{t_{sd}}{\mathbf{n_{sd}}} \right) / \; 2 \qquad (1)$$

where t_s and n_s are the number of the extracted tokens and the total number of terms extracted from the WS interface, respectively. t_{sd} is the number of extracted tokens of the description part, and n_{sd} is the total number of the extracted terms from the description part.

Similarly, we consider Web service annotations to extend the clarity index to provide a more specific metric to measure the understandability sub-characteristic. We measure it with the following formula:

$$\mathrm{Understandability}_{\mathrm{Index}} = \left(\frac{\sum_{i=0}^{n} t_s}{n_s} + \frac{N_{An}}{N_E + N_{At}} + \frac{t_{sd}}{n_{sd}} \right) / \; 3 \qquad (2)$$

N_{An} is the number of annotations, N_E is the number of items, and N_{At} is the number of attributes.

As the main focus of this part of work is the Web services with no description, we focus primarily on the Clarity Index because the annotation part will also be missing, also we did not consider the readability index for Web service description, which will be consider on further work.

2) WSDL Quality Index

Although our main research focus is finding a solution to address current poorly written WS descriptions or WSs with no description, we found that it is important to address the overall quality of the WS interface.

As explained in the related work Section, most previous studies focused on the WS interface structure without considering the understandability of the provided information inside. The quality of the WS interface is influenced by how much effort is required to understand its element types during either manual discovery or auto-discovery.

It is important to distinguish between two different aspects of interface quality. First, the complexity of the WS interfaces largely focuses on the interface structure. Second, the clarity and understandability largely focuses on understanding the Web service interface in terms of ease of reading and understanding the WS functionality. The combination of both can provide the required WS interface Quality control. To illustrate our idea, we propose a WSDLQuality Index that extends the current efforts in this area by including the defined Understandability metric. The same approach is valid for any other developed WSDL quality metrics.

$$WSDLQuality\ Index = \mathrm{Complexity}_{\ Index} + \mathrm{Understandabity}_{Index} \qquad (3)$$

In the evaluation Section, we explain the importance of the Clarity Index in more detail.

 ISBN: 978-1-61208-498-5

D. Sharing WSDL Quality Metrics

A WS interface document initially defines the methods of the service and how they are invoked, but it lacks support for nonfunctional properties such as QoS. The WSDL standard allows several powerful techniques for extending its schema to include or redefine elements and attributes [30]. The standard also allows organizing and structuring its schemas by breaking them into multiple files. These child schemas can then be included into a parent schema. Breaking schemas into multiple files has several advantages. First, it creates reusable definitions that can be used across several Web services. Second, it makes the definitions easier to read because it breaks down the schema into smaller units that are simpler to manage.

In this context, two main proposals have been introduced. First, a registry-based extension is proposed that focuses largely on the UDDI and use its tModels to express the extra required fields, which is a straightforward means of defining quality attributes in this type of Web service registry [31]-[33]. Second, a WSDL-based extension is proposed that uses a simple WSDL schema to provide required QoS information such as using annotations, model-driven techniques, or semantic concepts [34]-[36].

In this study, we considered a WSDL schema extension for Web service clarity and quality metrics sharing without addressing the transformation or mapping to a web registry structure because this is already covered by many approaches, as explained in the previous Section.

Fig. 5 and Fig. 6 depict the WS quality schema (WSQ) used to share the WS interface quality. It is worth noting that the main goal is not to provide a comprehensive catalog for the description of the WSDL quality attribute but to show how we can share such characteristics and metrics.

```
<WSQ>
    <ClarityIndex>0.25</ClarityIndex>
    <ComplexityIndex>0.39</ComplexityIndex>
    <TokensList> PAY Transaction MAC Transaction Transaction MAC MAC
Transaction Transaction Transaction MAC MAC MAC MAC</TokensList>
    <DistinctTokens> mac, pay, transaction</DistinctTokens>
    <DistinctTokensTF> 7,1,6 </DistinctTokensTF>
    <ArrayOfTokenTF>
        <Token>mac</Token>
        <Token>pay</Token>
        <Token>transaction</Token>
        <TokenTF>7</TokenTF>
        <TokenTF>1</TokenTF>
        <TokenTF>6</TokenTF>
      </ArrayOfTokenTF>
    <AvailabilityIndex>0.95</AvailabilityIndex>
</WSQ>
```

Figure 5. Example of a schema extension

Sharing the WSDL quality metrics has a number of advantages. First, the characteristic of the proposed WSDL follows the current approaches for WS QoS sharing, which can effectively be used to specify and share not only Web service interface quality control-related measures but also the SLA and service-provider trustworthiness metrics. Second, it improves the Web service reachability, composition, and usability by providing detailed information about the WS interface such as the distinct tokens and its term frequency (TF).

These tokens support both automatic syntactic discovery of the Web service and the Semantic discovery approach by providing a means that can help in identifying the Web service domain and classification. Third, it removes the burden of repeating the same processing every time during the automatic discovery phase by the service requestor. In general, it helps in resolving the missing WS description problem by providing an alternative means for extracting meaningful tokens to empower the missing description part. In addition, it provides the required control to measure the clarity and quality of the Web service interface to filter out any improper services.

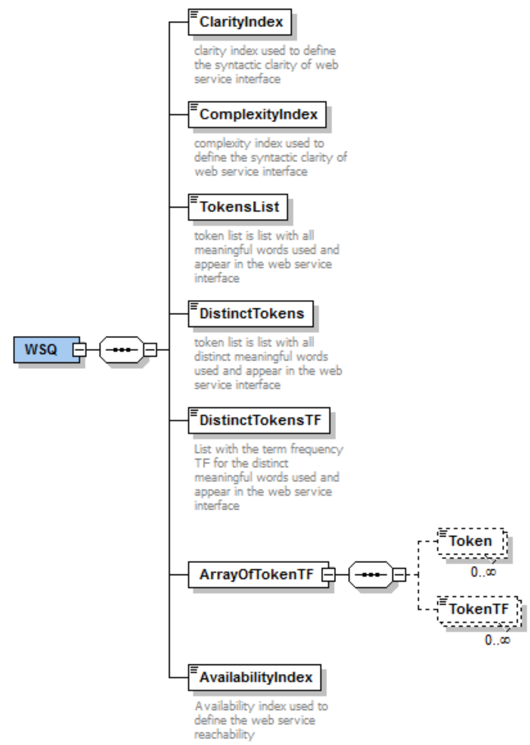

Figure 6. WSQ Schema Structure

VI. EXPERIMENTAL EVALUATION AND DISCUSSION

In this Section, we experimentally measure and compare the results of applying the Clarity Index to a list of Web services. In addition, we discuss the benefits of combining the complexity and clarity metrics to create a better WS interface Quality index.

The aim of this experiment is to show the capability of our approach to narrow the gap between Web services with and without a service description in terms of syntactic-based discoverability and usability. The results are summarized in Fig. 7, which shows the analysis of 10,000 active Web services, 52% of which have a Clarity Index greater than 40% and 79% of which have a Clarity Index greater than 30%. This shows that our approach can address and provide good support even for those Web services that have no description because it does not depend only on the description part but instead also relies on the attribute and element names used in the WS interface that we used to empower the description part.

Fig. 8 illustrates the Clarity Index similarity pattern of a Web service that has a description, and of the Web services

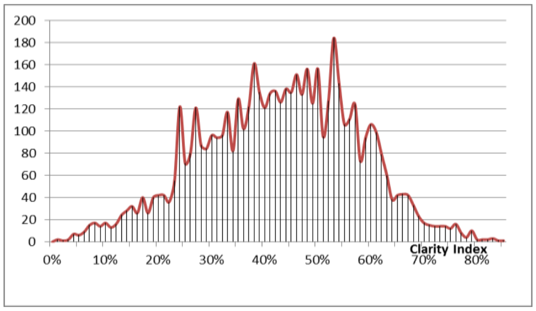

Figure 7. Clarity Index Distribution

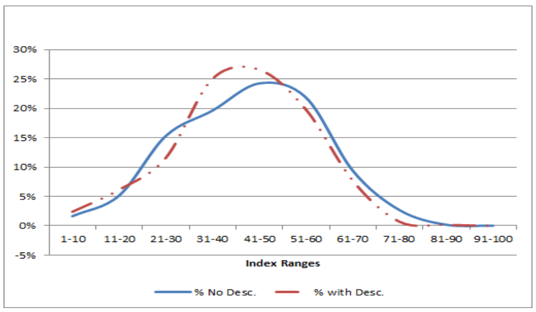

Figure 8. Clarity Index Pattern Comparisons

that do not have a description. The pattern matching reflects the understandability level of Web services, which allows the use of Web services with no description similarly to Web services with a description during the discovery process.

Note that in our previous analysis, 80% of the published Web services had no description. In other words, this number of Web services is out of the search and discovery scope. Our approach provides a means to recover this large number back into the search and the discovery process, which will be reflected also as better usability.

Fig. 9 shows the positive effect of considering all elements of the Web service interface against considering the description part only.

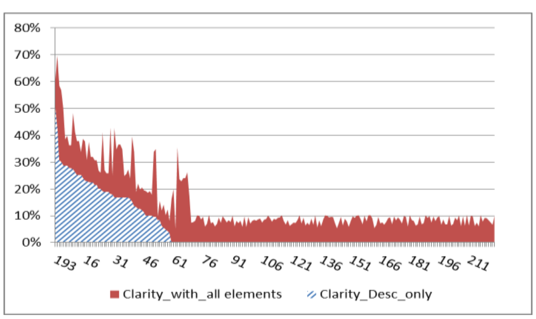

Figure 9. WS description Vs all elements Clarity Index

VII. CONCLUSION AND FUTURE WORK

In this study, our focus has been a practical problem of Web service discovery and usability, primarily WSDL-based Web services with poorly written descriptions. We developed a new approach to enhance the current Web service architecture capability that addresses the defined problem. Our approach is important for many reasons: first, it suggests an appropriate indicator of how Web service interface functionality is clearly described. This helps improve the automatic discovery and usability of Web services. Second, it helps in addressing the missing WS description problem by providing a technique for extracting meaningful tokens to empower the description missing part. Third, it provides a measure of interface clarity, which can be used during WS registration to filter out any unclear WS. Fourth, this approach is not limited to WSDL-based Web services but any other technologies as long a proper interface parser is available. In addition, we consider an approach to share the enhancement we introduce.

Our approach to share the WS interface quality metrics has a number of advantages. First, the proposed WSDL extension's characteristic follows the current approaches for WS QoS sharing. The proposed measure helps not only in WS interface quality control but also in the SLA and for service provider trustworthiness metrics. Second, it improves Web service reachability, composition, and usability by providing detailed information about the WS interface such as distinct tokens and term frequency (TF). These tokens support both the syntactic discovery of Web services and the semantic discovery approach by providing a mechanism that can help in identifying Web service classification. Third, the proposed approach helps in removing the burden of repeating the same processing every time during the discovery phase by the service requestor.

In this work, we focused mainly on Web service that has no description or a poor written description. For Web service with comprehended description, we are planning to consider the readability as a proxy for understandability and to extend the developed clarity and understandability indexes to include previously developed measurements as those used to study the readability of information including that returned by search engines. In addition, we are planning to study the advantage of having a common approach that consider different modern WS approaches such as JSON schema Apache Avro schemas, and other automated documentation generators.

REFERENCES

[1] S. Batra and S. Bawa, "Review of Machine Learning Approaches to Semantic Web Service Discovery," Journal of Advances in Information Technology, vol. 1, no. 3, pp. 146-151, 2010.

[2] M. Klusch and F. Kaufer, "WSMO-MX: A Logic Programming Based Hybrid Service Matchmaker," in European Conference on Web Services (ECOWS'06) , Zurich, 2006.

[3] E. Al-Masri and Q. H. and Mahmoud. [Online]. Available: http://www.uoguelph.ca/~qmahmoud/qws/. [Accessed March 2016].

[4] "Service-Repository," [Online]. Available: http://www.service-repository.com/. [Accessed Jan. 2016].

[5] J. Spillner. [Online]. Available: http://data.serviceplatform.org/. [Accessed Nov. 2015].

[6] U. Küster, "OPOSSum," [Online]. Available: http://fusion.cs.uni-jena.de/opossum/. [Accessed Jan. 2016].

[7] Z. Zheng and M. R. Lyu, "WS-Dream," [Online]. Available: http://www.wsdream.net. [Accessed Feb. 2016].

[8] M. Sabou, C. Wroe, C. Goble and G. Mishne, "Learning Domain Ontologies for Web Service Descriptions: An Experiment in Bioinformatics," in The 14th international conference on World Wide Web (WWW'05), Chiba, Japan, 2005.

[9] J. Fan and S. Kambhampati, "A snapshot of public Web Services," ACM SIGMOD Record, vol. 34, no. 1, pp. 24-32, 2005.

[10] C. Kiefer, A. Bernstein, H. J. Lee, M. Klein and M. Stocker, "Semantic Process Retrieval with iSPARQL," in The Semantic Web: Research and Applications, Innsbruck, Austria, Springer Berlin Heidelberg, 2007, pp. 609-623.

[11] J. Garofalakis, Y. Panagis and E. Sakkopoulos, "Web Service Discovery Mechanisms: Looking for a Needle in a Haystack?," in International Workshop on Web Engineering, Santa Cruz, 2004.

[12] E. Al-Masri and Q. H. Mahmoud, "Crawling Multiple UDDI Business Registries," in WWW '07 Proceedings of the 16th International Conference on World Wide Web, Banff, Alberta, Canada, 2007.

[13] L. Martin and T. Gottron, "Readability and the Web," Future Internet, vol. 4, no. 1, pp. 238-252, 2012.

[14] C. Fellbaum, WordNet An Electronic Lexical Database, Cambridge, MA ; London.: The MIT Press, 1998.

[15] A. McDowell, C. Schmidt and K.-b. Yue, "Analysis and Metrics of XML Schema," in Proceedings of the International Conference on Software Engineering Research and Practice, SERP '04, Las Vegas, Nevada, USA, 2004.

[16] M. H. Qureshi and M. H. Samadzadeh, "Determining the Complexity of XML Documents," in International Conference on Information Technology: Coding and Computing, ITCC 2005., Washington, DC, USA, 2005.

[17] Q. P. Thi, D. T. Quang and T. H. Quyet, "A Complexity Measure for Web Service," in KSE '09 Proceedings of the 2009 International Conference on Knowledge and Systems Engineering, Hanoi, Vietnam, 2009.

[18] D. Basci and S. Misra, "Measuring and Evaluating a Design Complexity Metric for XML Schema Documents," Journal of Information Science and Engineering, vol. 25, no. 5, pp. 1405-1425, 2009.

[19] M. Klettke, L. Schneider and A. Heuer, "Metrics for XML Document Collections," in Worshops XMLDM, MDDE, and YRWS on XML-Based Data Management and Multimedia Engineering — EDBT 2002, Berlin, 2002.

[20] R. Lammel, S. Kitsis and D. Remy, "Analysis of XML schema usage," in XML 2005 Conference, Atlanta, Georgia, USA, 2005.

[21] Y. Zhang, "Literature Review and Survey - XML Schema Metrics," University of Windsor, Ontario, Canada, 2008.

[22] J. Visser, "Structure Metrics for XML Schema," in XATA 2006, Portalegre, Portugal, 2006.

[23] H. M. Sneed, "Measuring Web Service interfaces," in 12th IEEE International Symposium on Web Systems,WSE 2010, Timisoara, Romania, 2010.

[24] U. Kumari and S. Upadhyaya, "An Interface Complexity Measure for Component-based Software Systems," International Journal of Computer Applications, vol. 36, no. 1, pp. 46-52, 2011.

[25] A. V. Paliwal, B. Shafiq, J. Vaidya, H. Xiong and N. Adam, "Semantics Based Automated Service Discovery," IEEE Transactions on Services Computing, vol. 5, no. 2, pp. 260-275, 2012.

[26] Z. Zheng, Y. Zhang and M. R. Lyu, "Distributed QoS Evaluation for Real-World Web Services," in IEEE International Conference on Web Services (ICWS 2010), Miami, FL, USA, 2010.

[27] M. Klusch and Z. Xing, "Deployed Semantic Services for the Common User of the Web: A Reality Check," in ICSC 2008, Santa Clara, CA, 2008.

[28] ISO/IEC, "ISO/IEC 25010:2011," [Online]. Available: http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=35733. [Accessed Dec. 2015].

[29] J. Spillner. [Online]. Available: http://data.serviceplatform.org/. [Accessed Nov. 2015].

[30] D. Waldt, "Six strategies for extending XML schemas in a single namespace," [Online]. Available: http://www.ibm.com/developerworks/library/x-xtendschema/. [Accessed Jan. 2016].

[31] A. Blum and F. Carter, "Representing Web Services Management Information in UDDI," 2004.

[32] Z. Xu, P. Martin, W. Powley and F. Zulkernine, "Reputation-Enhanced QoS-based Web Services Discovery," in IEEE International Conference on Web Services. ICWS 2007, Salt Lake City, UT, 2007.

[33] C.-C. Lo, D.-Y. Cheng, P.-C. Lin and K.-M. Chao, "A Study on Representation of QoS in UDDI for Web Services Composition," in Complex, Intelligent and Software Intensive Systems, CISIS 2008, Barcelona, 2008.

[34] A. D'Ambrogio, "A model-driven wsdl extension for describing the qos of Web services," in The IEEE International Conference on Web Servicess (ICWS'06), Chicago, USA, 2006.

[35] Y. Kang, "Extended Model Design for Quality Factor Based Web Service Management," in Future Generation Communication and Networking (Fgcn 2007), Jeju , 2007.

[36] K. Zhu, Z. Duan and J. Wang, "Quality of Service in Web Services Discovery," in Advanced Management of Information for Globalized Enterprises (AMIGE 2008), Tianjin, 2008.

[37] G. Zuccon, "Understandability Biased Evaluation for Information Retrieval," in Advances in Information Retrieval, Springer International Publishing, 2016, pp. 280-292.