WSDLUD: A Metric to Measure the Understanding Degree of WSDL Descriptions

Mario Marcelo Berón[1(✉)], Hernán Bernardis[1], Enrique Alfredo Miranda[1], Daniel Edgardo Riesco[1], Maria João Varanda Pereira[2], and Pedro Rangel Henriques[3]

[1] Department of Computer Science, Universidad Nacional de San Luis, San Luis, Argentina
{mberon,hbernardis,eamiranda,driesco}@unsl.edu.ar
[2] Centro Algoritmi, Universidade do Minho, Instituto Politécnico de Bragança, Bragança, Portugal
mjoao@ipb.pt
[3] Department of Computer Science/Centro Algoritmi, University of Minho, Campus de Gualtar, Braga, Portugal
prh@di.uminho.pt

Abstract. In this article, WSDL Understanding Degree (WSDLUD) a metric aimed at measuring a priori the understandability of WSDL (Web Services Description Language) descriptions is presented. In order to compute WSDLUD, all the static information available in a WSDL description is collected. This information is submitted to an evaluation process based on a method named LSP (Logic Scoring of Preference). This evaluation process outputs a Global Preference value that indicates the satisfaction level of the WSDL description regarding the evaluation focus, in this case, the understanding degree.

Keywords: WSDL · Web services comprehension · LSP

1 Introduction

Nowadays the Web Services (WS) are fundamental software artifacts for building service oriented applications. According to World Wide Web Consortium (W3C, for details see http://www.w3.org/), a WS is: *a software application identified by a URI, whose interfaces and bindings are capable of being defined, described, and discovered as XML artifacts. A WS supports direct interactions with other software agents using XML-based messages exchanged via Internet-based protocols.* The organizations, increasingly, produce web services which are used by other organizations to produce new software systems aimed at solving business demands. Web services have associated a description which specifies the data types used, the operations provided, inputs and output, the technology used to accomplish the communications between other high level and low level

J.-L. Sierra-Rodríguez et al. (Eds.): SLATE 2015, CCIS 563, pp. 91–100, 2015.
DOI: 10.1007/978-3-319-27653-3_9

of software elements. These descriptions are published in the internet and the organizations can retrieve them and decide if some of those services are useful for building the software they need [12]. Web Services are software packages and therefore they must be comprehend for maintenance tasks (bug fixing, adaptation, evolution, etc.). The primary information source to accomplish this task is the respective WSDL (Web Service Description Language, http://www.w3.org/TR/wsdl20/) description. Although, there are several resources from which it is possible to collect information about the Web Service, the WSDL description is the first that the user employs for analysing its usefulness for his purposes. Furthermore, the web service descriptions are interesting because they provide a high level abstraction data which can be very useful to simplify the understanding of the web services. As said above, a standard language used to write web service descriptions is WSDL. This language is a dialect of XML with well defined rules to specify each component. Being a XML based language it is fastidious to read such a description, and therefore a tool is needed to assist the software engineer in this task. In this context, many tools can be found that are oriented to facilitate the inspection of WSDL descriptions, transform to a different WSDL version, compute several metrics, produce user-friendly visualizations, etc. However, at the best of our knowledge, only a few are oriented to help their understanding. Taking this into consideration, in this article WSDLUD (Web Service Understanding Degree) is presented. WSDLUD is a metric aimed at providing, a priori, a measurement about the WSDL description understanding complexity. For calculating WSDLUD, Logic Scoring of Preference Method (LSP) [7,14] is used. LSP is a multi-criteria evaluation method; it requires a Criteria Tree, an Aggregation Structure and a set of Elementary Criteria Functions to be defined. Combining systematically such elements, this method produces a satisfaction level that indicates, in this case, the understanding degree of a WSDL description. In order to apply LSP and compute WSDLUD, the WSDL description must be statically analysed and all the information available must be retrieved. This information is submitted to different evaluation procedures in order to obtain satisfaction values (values in [0,1] or [0,100]). To perform these processes, the use of both compilation and natural language processing techniques are required. The first is used to retrieve formal elements from WSDL source code. The second is employed to gather semantic information from unstructured information sources.

The article is organized as follow. Section 2 describes the work tightly related with the research topics here presented. Section 3 defines the WSDLUD evaluation structures. Section 4 presents the case studies where it is possible to observe the results obtained through the application of WSDLUD to some test cases available in W3C. Section 5 closes the paper with some conclusions and future work.

2 Related Work

The WSDL description analysis is based on static and behavioral information. The traditional approaches are oriented to compute metrics to compare and evaluate a set of program parameters [2,13,15].

Considering static information, authors [13] have defined metrics for organization security. In this context, the authors affirm that the easier to understand a WSDL description the easier will be to carry out fraudulent actions against the organization. On account of that, the authors compute the understanding level of WSDL description and if it is high they define approaches to diminish its readability.

The second, based on behavioral information, is concerned with measuring the WSDL description considering the complexity of the operations and messages involved. The more complex the operations and messages are, the more complex will be to understand the WSDL description [9].

It is also possible to find works that use ad-hoc approaches. They are based on traditional object oriented metrics to measure quality attributes of WSDL descriptions [5,6].

WSDLUD metric, defined in this article, is different from those found in the literature in several aspects. First, all formal elements of the WSDL description (types, port types, bindings, services) are considered and for each one of them the understanding degree is measured.

Second, the WSDL description's understandability can be simplified if the informal information (those provided by the identifiers and documentation) gives useful semantic information about the description's domain. For this reason, several metrics to measure the quality of the identifiers and documentation of the description, are defined and calculated.

Third, the value of our metric is produced by the combination of other metrics (those mentioned before) which consider both formal and informal information. We used these metrics to measure WSDL descriptions and obtain a final value for each of them. This final value is computed by using a multi criteria method. This method is parameterizable allowing to reflect the engineer experience in the evaluation mechanism. Finally, as a side effect, the process used to compute WSDLUD can also be used for: (i) To provide a ranking of WSDL descriptions understandability, (ii) To build visualizations based in charts, and allow to analyse the results and to discover the possibilities to improve the WSDL description understanding.

To finish this section, it is important to notice that, at best of our knowledge, a metric with the characteristics mentioned above was not described in the literature. So, we believe that the work here reported is a valid contribution for the comprehension of WSDL specifications.

3 WSDLUD

In this section, all the concepts and processes involved in the definition and measurement of WSDLUD are described in detail.

3.1 WSDL Description Criteria Tree

The criteria tree of a WSDL description (these characteristics were extracted from a WSDL specification provided by W3C.) is composed by the following

characteristics: (i) Type Understanding Degree, (ii) Message Understanding Degree, (iii) Port Type Understanding Degree, (iv) Binding Understanding Degree and (v) Service Understanding Degree. Each characteristic has an associated sub criteria tree which takes into consideration the proper properties of the evaluated element.

In the next paragraphs the characteristics mentioned above will be developed, for each of them, the Criteria Tree will be explained.

Type Understanding Degree. This characteristic is composed by the following attributes: *Number of Primitive Types*, *Number of Complex Types*, *Documentation Quality*, *Type Name Quality* and *Number of Fields*. Clearly, a primitive type (a primitive type is a type provided by the language), for example: text, integer, real, boolean, etc. will be easier to understand than a complex type (a complex type is a type defined by the user). A primitive type can be deduced from its identifier and the explanations provided by the language manual. A complex type is more difficult of perceiving because it is composed by several identifiers, which are susceptible to do many analysis and the explanations exposed in the language manual are not enough. In this context, if the documentation provided is bad or null, the comprehension will be even more difficult.

Message Understanding Degree. This characteristic can be evaluated taking into consideration the following attributes: *Message Documentation Quality*, *Message Name Quality* and *Part Understanding Degree*. Concerning the first two elements, it is possible to say that they will provide relevant information when some semantic information can be extracted. For that the following components are considered: name, element name and type. The sub-characteristic named *Part Understanding Degree* which can be divided in *Part Name Quality*, *Part Element Name Quality* and *Part Type Understanding Degree* attributes. All these attributes must also be considered when the message understandability needs to be measured.

Port Type Understanding Degree. This characteristic has the following attributes: *Port Type Name Quality*, *Port Type Documentation Quality* and *Port Operation Understanding Degree*.

The first two are important because they provide semantic information when they are well defined. Semantic information can also be extracted from *Port Operation Understanding Degree* measuring the *Port Type Operation Understanding Degree*.

The definition of this characteristic follows the same approach that message part, in other words to each simple operation element we consider some attributes like name, documentation, parameters, etc. (more details about the disaggregation of this sub-characteristic can be found in [3]).

Binding Understanding Degree. This characteristic is composed by the following attributes: *Binding Name Quality*, *Binding Documentation Quality*, *Binding Type Understanding Degree* and *Binding Operation Understanding Degree*.

Once more the name quality and the documentation quality are important characteristics to measure using the attributes: *Binding Name Quality* and *Binding Documentation Quality*. The other two attributes are already defined in others characteristics. *Binding Type Understanding Degree* is defined in *Type Understanding Degree* and *Binding Operation Understanding Degree* is defined in *Port Type Understanding Degree*. For this reason, during evaluation process we re-use the values obtained in previous computation.

Service Understanding Degree. A service is made available by a WSDL description. A service has a name and documentation and it is composed by ports. For analyzing the *Service Understanding Degree* it is necessary to measure *Service Name Quality*, *Service Documentation Quality* and *Service Port Understanding Degree* in a Service context.

3.2 Aggregation Structure

As LSP method states [14], the satisfaction values that result from the application of the Elementary Criteria Functions to the measurable attributes, must be aggregated in order to obtain the Global Preference. This Global Preference represents the satisfaction of the object under evaluation. As could be seen in Subsect. 3.1, we propose a Criteria Tree for each WSDL element (type, message, port, etc.). For each of these Criteria Trees, we developed a specific Aggregation Structure. To illustrate the approach and to save space, in Fig. 1 we only show the Aggregation Structure for the characteristic *Message Understanding Degree*.

We used a partial absorption LSP function (compound by operator A (arithmetic mean) and SQU (square mean) — all the LSP operators are better explained in [8]) to aggregate Message Documentation Quality and Message Name Quality. This kind of asymentric compound operators are used when some input values could be zero (non-mandatory input). It is necessary because in many cases, messages do not have a good documentation (sometimes do not have at all). A medium conjunctive operator (CA) is used to compute the Message Understanding Degree Global Preference. This kind of operator is employed when the input requirements are mandatory. Thus if one of the input values is zero, the operation result will be zero. The weights are used to express the relative importance of input preference. As message documentation and name provides more significant semantic information, its weight is 70 %, as opposed to *Part Understanding Degree* which provides less semantic information (its weight is 30 %).

3.3 Information Extraction Techniques and Elementary Criteria Functions

The information extraction techniques and the Elementary Criteria Functions are the most important features for the evaluation process that will be described.

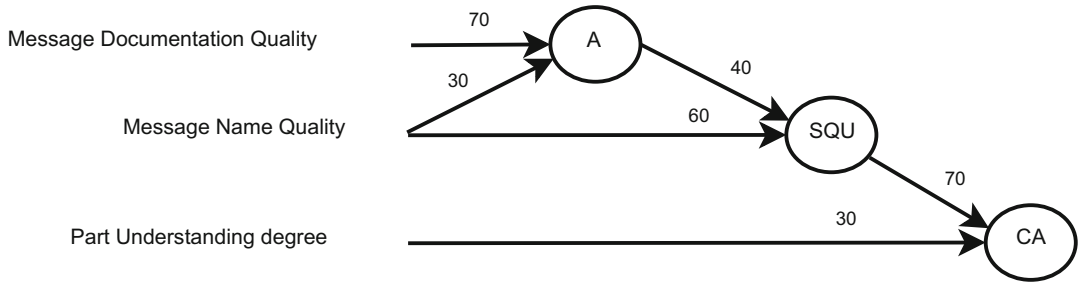

Fig. 1. Message understanding degree Aggregation Structure.

The former allows to obtain the information and perform all the analysis to get each attribute value for the Criteria Tree. The latter maps each of these in a satisfaction level, i.e., a value in the interval [0,1] (or [0,100]). This value represents the satisfaction degree of the attribute for the object under evaluation according to the sensibility and experience of the authors.

Information Extraction Techniques. The approach used to extract information from a WSDL description combines compilation techniques, natural language processing algorithms and strategies to compute indicators [4]. The first are implemented using DOM (Domain Object Model) a parser for XML language which explicitly builds an internal representation of the analysed XML source code. Several traversals are applied through this internal representation for gathering the desired information. The identifiers and the documentation are extracted by using compilation techniques. In order to retrieve semantic information IdA (Identifier Analysis) [1] is used. IdA is a tool aimed at applying algorithms to divide, expand and find a meaning for the identifiers of a program. Finally, with the goal to provide a measure about of the understanding degree of a WSDL description, NESSy [11] was used. NESSy is a tool to evaluate software based on LSP method.

For attributes like *Type Name Quality* (see in Algorithm 1 the computation process of the satisfaction level of Type Name Quality Criterion), *Message Name Quality* or *Binding Name Quality* we use identifier analysis techniques.

The purpose of this analysis is to discover the relation between the names and the concepts of the problem domain. The name quality is higher when its related words are meaningful. The result of the techniques is a percentage which indicates the satisfaction level for a particular name quality.

For attributes like *Type Documentation Quality*, *Message Documentation Quality*, *Binding Documentation Quality*, etc., we use documentation analysis techniques. This kind of attributes has as main goal to measure the usefulness level of the information provided by the element's documentation (IdA also is used to carry out this task). The analysis techniques gathers documentation and returns a percentage which represents the satisfaction level for the attribute under study. In first place the documentation is divided by words, then the irrelevant words are deleted. The next step consists of analysing each word and

count those that have a useful meaning. The result is obtained carrying out the following computation: $\frac{Number\ of\ Word\ with\ Mean}{Number\ of\ Words}$.

Algorithm 1. Satisfaction Level of Type Name Quality Criterion

```
input : typeName a string which represents a type name.
output: Satisfaction Level, a percentage that indicates the
        satisfaction level of the criterion Type Name
        Quality.
Data: wordSet,stopWords a set of words.
Data: pal a string which represent a word extracted from a
      type name.
Data: wordsWithMeans an integer variable which counts the
      number of words extracted from a type name which have
      meaning.
wordSet←division(typeName);
stopWords←extractStopWords(wordSet);
wordSet←wordSet-stopWords;
wordsWithMean←0;
foreach w in wordSet do
    pal←expand(w);
    if hasMean(pal) then
        wordsWithMean←wordsWithMeans + 1;
end
return (wordsWithMeans / |wordSet|);
```

Elementary Criteria Functions. In this evaluation process, the majority of Elementary Criterion Function are direct mappings, since most of the attributes values are computed by extraction techniques. They take as input the strings to be analysed and return a percentage value that could directly be mapped to a satisfaction value.

4 Case Study

This section presents the evaluation of five WSDL descriptions using LSP and the structures defined in Sect. 3 [10]. All descriptions belong to web services frequently used by information systems:

(i) *Google Web APIs* (https://code.google.com/p/dic/downloads/detail?name=GoogleSearch.wsdl), provides operations to do Google searchs, (ii) *Create Queue (Amazon)* (http://queue.amazonaws.com/doc/2009-02-01/QueueService.wsdl), offers a reliable, highly scalable hosted queue for storing messages as they travel between computers, (iii) *Airport* (http://www.webservicex.com/airport.asmx?wsdl), provides useful information of all world airports (e.g. airport codes, names, countries, countries code, latitude, longitude, etc.) (iv) *Global Weather* (http://wsf.cdyne.com/WeatherWS/Weather.asmx?WSDL), gets weather report for all major cities around the world, and

(v) *OFAC* (http://www.webservicex.net/OFACSDN.asmx?WSDL) aids banks in meeting the requirements of the US Treasury Department's Office of Foreign Asset Control (OFAC).

Table 1. Partial and global evaluation of WSDL

High-Level characteristic	Google	Weather	Amazon	Airport	OFAC
Types U. D	60,2665	71,5131	68,8148	72,2303	40,5846
Messages U. D	69,1173	83,3624	79,753	77,4924	58,8801
Port Types U. D	75,7194	81,4166	82,1289	81,8902	45,3519
Bindings U. D	75,5258	79,3457	82,2505	79,5241	42,755
Services U. D	78,9946	79,6724	89,4138	79,7011	42,0794
Final Scores	71,5594	77,0112	80,1496	78,0884	45,4495

Table 1 shows the global understanding degree for each WSDL description. Each *Global Preference* was computed aggregating all the characteristic preferences with the logical operator *CA* (this function simulates simultaneity) and the weight equally distributed among the characteristics (20 % for each one). The choice of this operator is due to the fact that all WSDL components (type, message, port type, etc.) must be understandable. If one of these is incomprehensible, the whole WSDL will be difficult to understand.

As can be seen in Table 1, almost all WSDL are very similar taking into account understanding degree, except for *OFAC* WSDL description. This is because that description has numerous identifiers with acronyms which decreases the satisfaction levels.

Weather and *Airport* define each type using a few primitive and complex types. Furthermore they specify explicit and unambiguos identifiers. On the other hand, *Google* uses a number of primitive and complex types that exceed the established thresholds. The majority of messages's parts of *Weather* WSDL uses primitive types and this fact rise its *Messages Understanding Degree* satisfaction value.

In general, *Amazon* WSLD presents more documentation than others in different parts, like messages, types, port types and services. This makes this WSDL the most understandable of the case study.

From another point of view, this set of metrics was proposed to measure each component individually inside a WSDL. In this sense, we could compare, for example, all elements of a kind that a WSDL contains (e.g. types, messages or services), in order to analyze it individually. This is could be useful for maintainability or re-structuring purposes. In this context, we measure three messages that presents *Create Queue (Amazon)* WSDL description. In this context, we measure the quality of three different messages of the Create Queue (Amazon) WSDL description and the results can be seen in Table 2.

As can be seen, *RemovePermissionRequest* (RPR) message is the most understandable of these three messages and *SendMessageResponse* (SMR) the worst. This is basically due to *Message Part Understanding Degree* satisfaction values.

Table 2. Messages individual measurement of *Create Queue (Amazon)* WSDL description.

Sub-characteristic	SendMessage Response	RemovePermission Request	DeleteMessage Response
M. Doc. Quality	0	0	0
M. Name Quality	100	100	100
M. Parts U. D	60,9759	93,6933	73,1726
Final Scores	73,17	83,5379	77,6729

This is a comparative analyse that allows to identify the most critical parts of the description. If we want to analyse the results individually we would say that a score less than 50 % represents a candidate description for improvement.

5 Conclusion and Future Work

In this article WSDLUD a metric, to measure the understanding degree of WSDL description, was defined. In order to compute WSDLUD other metrics were also specified. These metrics have as main goal to provide an estimation about the understanding degree of each description part. Each part is associated with an importance level specified by the engineer. Both values (understanding degree and importance level) are used by LSP (a multi criteria evaluation method) to produce a global value which represents the desired WSDL description understanding degree.

We believe that our approach is novel because it makes possible to analyse each part of a particular WSDL description as well as the global understanding degree. Yet more important, all the engineer's experience can be included in the evaluation process in order to get more significant results. All the detailed information provided by our system can be used to identify the most critical parts of the description and the chances for quality improvement. In some cases, the description can be simplified or made more readable. But, in other cases, the complexity of the description is full dependent on the domain complexity and there is not chance for improvement.

As future work we intend to:

(i) Improve the Criteria Tree (CT) and Aggregation Structure (AS);
(ii) Extend the work presented in this paper to WSDL 2.0;
(iii) Apply a similar analysis to study business processes specified with BPEL (Business Process Execution Language).

Acknowledgements. This work has been supported by FCT–Fundação para a Ciência e Tecnologia within the Project Scope: UID/CEC/00319/2013.

References

1. Azcurra, J., Berón, M., Montenjano, G., Farnese, A., Henriques, P., Pereira, M.: AId: Uma Ferramenta para Análise de Identificadores de Programas Java. In: Congreso Nacional de Ingeniería Informática/Sistemas de Información, pp. 880–892, Noviembre 2014
2. Bernardis, H., Beron, M., Riesco, D., Henriques, P.R.: Extracción de información y cálculo de métricas en WSDL 1.1 y 2.0. In: Congreso Nacional de Ingeniería Informática/Sistemas de Información, pp. 963–974, Noviembre 2014
3. Beron, M., Henriques, P.R., Riesco, D., Pereira, M.J.V.: On the Comprehension of WSBPEL Programs. Technical report, Universidad Nacional de San Luis - Universidade do Minho (2015)
4. Carvalho, N.R.: An Ontology Toolkit for Problem Domain Concept Loction in Program Comprehension. Ph.D. thesis, Escola de Engenaria, Universidade do Minho (2014)
5. Coscia, L.O., Crasso, M., Mateos, C., Zunino, A.: Estimating Web Service interface quality through conventional object-oriented metrics. CLEI Electron. J. **16**(1) (2013)
6. Coscia, L.O., Mateos, C., Crasso, M., Zunino, A.: Refactoring code-first Web Services for early avoiding WSDL anti-patterns: approach and comprehensive assessment. Sci. Comput. Program. **89**, 374–407 (2014)
7. Dujmovic, J.: Continuous preference logic for system evaluation. IEEE Trans. Fuzzy Syst. **15**(6), 1082–1099 (2007)
8. Dujmovic, J.: Characteristic forms of generalized conjunction/disjunction. In: IEEE International Conference on Fuzzy Systems, FUZZ-IEEE 2008, (IEEE World Congress on Computational Intelligence), pp. 1075–1080. IEEE (2008)
9. Kumar, R., Indraveni, K., Goel, A.K.: Automation of detection of security vulnerabilities in Web Services using dynamic analysis. In: 9th International Conference on Internet Technology and Secured Transactions (ICITST), pp. 334–336, December 2014
10. Liu, L., Sun, T., Fang, W., Liu, N.: Usability evaluation of the subway train dispatching system. In: 2011 International Conference on Information Science and Technology (ICIST), pp. 1123–1128, March 2011
11. Miranda, E., Berón, M., Montejano, G., Pereira, M.J.V., Henriques, P.R.: NESSy: a New Evaluator for Software Development Tools. In: 2nd Symposium on Languages, Applications and Technologies, SLATE 2013, Porto, Portugal, pp. 21–37, 20–21 June 2013
12. Newcomer, E.: Understanding Web Services: XML, WSDL, SOAP, and UDDI. Addison-Wesley Professional, New York (2002)
13. Sripairojthikoon, P., Senivongse, T.: Concept-based readability measurement and adjustment for web services descriptions. In: 16th International Conference on Advanced Communication Technology (ICACT), pp. 378–388, February 2014
14. Su, S., Dujmovic, J., Batory, D.S., Navathe, S.B., Elnicki, R.: A cost-benefit decision model: analysis, comparison and selection of data management. ACM Trans. Database Syst. **12**(3), 472–520 (1987)
15. Tibermacine, O., Tibermacine, C., Cherif, F.: A Practical Approach to the Measurement of Similarity between WSDL-basedWeb Services. RNTI: Revue des Nouvelles Technologies de l'Information, Special Issue CAL 2013 (RNTI-L-7): 03–18 (2014)