

Enlarging Test Collections of Composite Semantic Services

Hadi Saboohi, Amineh Amini, Tutut Herawan, Sameem Abdul Kareem, Nooshin Anari, and Gholamreza Ahakian

Abstract—There are numerous methods which are proposed to mediate semantic Web services. The mediations of semantic services include their discovery, composition, execution, and monitoring. These are performed on both atomic and composite services. Newly proposed methods are required to be evaluated on a dataset. Despite the existence of atomic test collections of semantic services, the number of publicly available test collections containing composite services is not comparable. In this paper, we propose an approach to enlarge the number of composite services in a test collection. We generate new composites by calculating subdigraphs of available composite services. We evaluated our approach on a number of composite services and we could exponentially enlarge the size of test collections.

Index Terms—Semantic web service, composite service, test collection.

I. INTRODUCTION

Academicians are carrying out various researches on Web services because numerous potentials exist in the nature of Web services [1]. In recent years, the use of Web services has grown rapidly due to the introduction of the “Semantic Web.” Semantic annotations of Web services enable agents and in general machines to autonomously interact with each other [2].

Similar to any research area, new methods need to be evaluated, and the evaluations require a real set of data. There are thousands of traditional Web services with their descriptions on the Web. However, when it comes to semantic Web services, the number of publicly available ones is dramatically few and they are not bundled in a test collection. Furthermore, most of the existing semantic Web services are not created from real Web services.

Some research groups created test collections of semantic Web services. The test collections are meant to be used for a specific research or to be published as a test suite for the research community. Current available test collections of

semantic Web services are SWS-TC [3], OWLS-TC [4], SAWSDL-TC [5], and WSMO-Lite-TC [6]. The first two test collections are created using the OWL-S [7], [8], SAWSDL-TC is created for the SAWSDL [9], and the latter is in WSMO-Lite [10]. SWS-TC includes semantic descriptions of 241 services, whereas OWLS-TC has 1083 services and both SAWSDL-TC, and WSMO-Lite-TC collections include 1080 services. Various characteristics of public test collections are investigated in [11], [12].

Table I, which is adapted from the paper presented in [12], shows current available test collections of semantic Web services, i.e. SWS-TC, OWLS-TC, SAWSDL-TC, and WSMO-Lite-TC. It shows the semantic Web service description language in which the services of the test collections are described, number of their services, and their latest release year. All these test collections include only atomic services.

One major reported problem shared among the test collections is that none of them contains composite Web services. Nevertheless, mediation techniques need composite services to evaluate the works.

In this paper, we propose a method in order to generate new composite services based on the subdigraph calculations of available composite services. Hence, it enlarges the size of a composite semantic Web services’ test collection.

The rest of the paper is organized as follows. Related works are investigated in the next section. Section III shows the way we represent our graphs and describes our subgraph calculation method. Discussion is given in Section IV and the section concludes our paper as well.

TABLE I: CURRENT TEST COLLECTIONS OF SEMANTIC WEB SERVICES (ADAPTED FROM [12])

Test Collection	Semantic Web Service Description Language	Number of Services	Last Release Year
SWS-TC	OWL-S	241	2006
OWLS-TC	OWL-S	1083	2010
SAWSDL-TC	SAWSDL	1080	2010
WSMO-Lite-TC	WSMO-Lite	1080	2012

SWS-TC [3]: Semantic Web Service Test Collection
 OWLS-TC [4]: OWL-S Service Retrieval Test Collection
 SAWSDL-TC [5]: SAWSDL Service Retrieval Test Collection
 WSMO-Lite-TC [6]: WSMO-Lite Service Test Collection

II. RELATED WORK

In the literature, most of the researches on semantic Web services are usually evaluated on a small number of services (such as [13]-[16]).

Manuscript received July 15, 2014; revised September 22, 2014.

Hadi Saboohi, Amineh Amini, Tutut Herawan, and Sameem Abdul Kareem are with the Faculty of Computer Science and Information Technology, University of Malaya (corresponding author; e-mail: saboohi@um.edu.my).

Nooshin Anari is with the Department of Computer, Kerman Islamic Azad University, Kerman, Iran.

Gholamreza Ahakian is with the Department of Computer, Varamin-Pishva Islamic Azad University, Pishva, Iran.

There is an online portal for semantic services called OPOSSum [17], which assembles data from SWS-TC, OWLS-TC and some other sources. It creates an assemblage of over 1500 semantic Web services using over 2800 descriptions for different description languages [11]. Unfortunately, the portal is not updated anymore and newly improved releases of the test collections are not imported into the portal.

Recently, the characteristics of the available test collections in term of their capabilities for world-altering services are investigated in [12]. None of the investigated test collections include composite Web services.

There is an ongoing effort [18], [6] in making a new test collection of semantic Web services at SEALS Community [19]. The test collection is named WSMO-LITE-TC and it is based on WSMO-Lite [10]. WSMO-Lite is the most recent submission to W3C as a semantic Web services' description language. However, according to the primary report, this new test collection will not provide composite services as well.

Consequently, to the best of our knowledge there is no effort in making a test collection of composite semantic Web services.

III. DIGRAPHS OF COMPOSITE SERVICES

Composite services are modelled as directed graphs (digraphs [20]). The services make the vertices (nodes) and their relations, i.e. inputs, outputs, preconditions and results, make the edges. As a means of representing the digraphs of composite semantic Web services, i.e. representing which vertices are adjacent to which other vertices, we use the adjacency matrix. We use the concept names as the values of the matrix.

Another matrix representing a graph is the incidence matrix. We do not use the incidence matrix because we use the concept names (their codes) as the values of the adjacency matrix. These values need to be repeated if we want to use the incidence matrix which causes redundancy. Furthermore, because the usual *sizes* of our graphs are greater than their *order*, we prefer to use the adjacency matrix rather than incidence matrix.

	User(I)	WS ₁	WS ₂	WS ₃	WS ₄	User(O)
User(I)		I ₁	I ₂	I ₃	I ₄	
WS ₁			C ₁	C ₂	C ₃	O ₁
WS ₂		C ₁		C ₄	C ₅	O ₂
WS ₃		C ₂	C ₄		C ₆	O ₃
WS ₄		C ₃	C ₅	C ₆		O ₄
User(O)						

Fig. 1. Adjacency matrix of a valid digraph of a composite web service. coordinates are user inputs, constituent services (WS1 ..WS4), and user outputs.

An alternative for adjacency matrix is the adjacency list. However, because mainly needed operations on composite services, i.e. their mediation, require checking the existence

of an edge between two given vertices in no time, we chose to use the adjacency matrix rather than the adjacency list.

The coordinates of an adjacency matrix of our digraphs are User(I), the constituent Web services' names, and User(O). User(I) and User(O) identify the user of composite service who provides the inputs and receives the outputs of the (successful) execution. A non-diagonal entry is the concept used as the label for an edge between two Web services.

Fig. 1 shows a template of an adjacency matrix for the inputs, and the outputs of a composite Web service with four constitutive Web services (WS₁...WS₄). Every cell shows an edge from a vertex shown at the left coordinate to a vertex shown at the top coordinate.

A. Subdigraph Calculation

We calculate subdigraphs of a composite semantic Web service by a sequence of vertex- and edge-deletion based on the mathematical concepts of a digraph [20].

Algorithm 1, called "Test Collection Enlarger Recursive", shows a pseudo code of the proposed subdigraph calculation approach which aims to enlarge a test collection of composite services. It receives a composite service and returns a set of composite services.

Test Collection Enlarger Recursive is a recursive algorithm. It removes one of the vertices of the digraph, i.e. an atomic service from the composite. Then, it fixes the edges and adds the resulting service to its return list. It recursively calls itself with the newly generated subdigraphs. The recursive calls continue until it reaches its base case in which the received service is not a composite service anymore.

Algorithm 1 commences by checking if the received service is a composite one as its base case (Line 2). It checks the number of services in the composite using a predefined function called "Number Of Services." If the received (input) service is an atomic one, it returns an empty set and it means that there is no smaller service in the received input. After this checking, it removes every vertex, i.e. atomic service, through a loop (Lines 5-23). Line 6 removes one atomic service from the composite. Lines 7 to 12 either convert the 'in-edges' (input edges) of the removed atomic service to the output edges of the resulting service or remove them if they are internal edges. Similarly, lines 13 to 18 either convert the 'out-edges' (output edges) of the removed atomic service to the inputs of the resulting service or remove them if they are internal edges. Finally, if the newly generated service is not already in the list it will be added to the result set and the algorithm itself is called once more with this new service (Lines 19-22).

As an example, Fig. 2 shows a sample sequential composite semantic Web service with four smaller services.

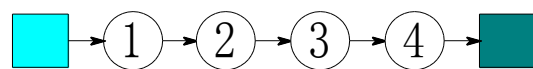


Fig. 2. A sample sequential composite semantic web service.

Moreover, Fig. 3 illustrates the subdigraphs of the sequential composite service in Fig. 2. Through the stages of the subdigraph calculation, we make some new services and we add them to an ordered list. Then, we process the newly added digraphs to generate new subdigraphs of services so as

to make more digraphs of composite services. We continue processing the ordered list until no more digraphs are added.

The stopping criterion is when we reach the atomic services of the digraphs.

```

Algorithm 1. Test Collection Enlarger Recursive(CSWS)

Input: CSWS, A Digraph of a Composite Semantic Web Service
Output: A Set of Subdigraphs of Composite Semantic Web Services

1: Result ← {};

   /* Checking for the base case */
2: If Number Of Services(CSWS) is 1 then
3:   Return Result;
4: End if;

   /* Removing every atomic service from the composite */
5: For all vertices, ASWS, in CSWS
6:   NewCSWS ← CSWS - ASWS; // Remove ASWS from CSWS

   /* Converting inputs of ASWS */
7:   For all in-edges of ASWS
8:     If the in-edge is the input of NewCSWS then
9:       Remove it;
10:    Else if
11:      Convert it to the output of NewCSWS;
12:    End for;

   /* Convert outputs of ASWS */
13:   For all out-edges of ASWS
14:     If the out-edge is the output of NewCSWS then
15:       Remove it;
16:     Else if
17:       Convert it to the input of NewCSWS;
18:     End for;

   /* Avoiding repeated composites in the result set */
19:   If NewCSWS is not in Result then
20:     Result ← Result + NewCSWS;

   /* Recursion Call */
21:   Result ← Result + Test Collection Enlarger Recursive(NewCSWS);
22:   End if;
23: End for;

   /* Returning final unique results containing some composite services */
24: Return Result;
    
```

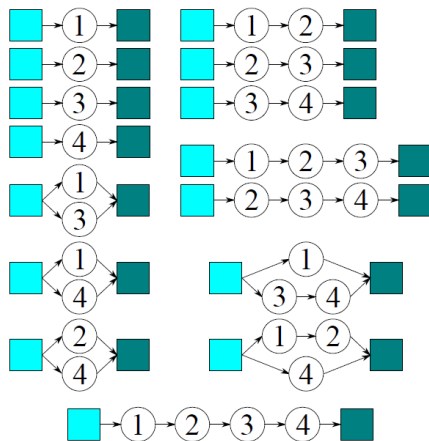


Fig. 3. All possible subdigraphs.

B. Implementation

OWLS-TC is a well-known and often-used test collection of atomic semantic Web services. The test collection has been used in S3 (Semantic Service Selection) contests¹.

¹ Annual International Contest S3 on Semantic Service Selection - Retrieval Performance Evaluation of Matchmakers for Semantic Web Services, <http://www-ags.dfki.uni-sb.de/~klusch/s3/>

OWLS-TC is a service retrieval test collection in which the atomic services are described in OWL-S language. OWLS-TC Version 4 contains 1083 atomic semantic Web services which are described using their Input and Output, and Precondition and Results (IOPRs).

We extracted the properties of atomic Web services from OWLS-TC². The properties include the service names (from human-readable information in Service Profile) along with their functional properties. Moreover, the concepts used in the services are extracted from their related ontologies available in the collection.

Then, we generated some synthetic composite semantic Web services using the atomic services of OWLS-TC. Finally, we calculated all the subdigraphs of the composites. The proposed method could add many new composite services; hence, the size of the collection has increased.

The structures of the generated composites can be translated and exported to the needed semantic Web services languages such as OWL-S. Moreover, their adjacency matrices are exported as well. A set of composites generated were used in some researches [21].

² The extraction was performed using OWL-S API

IV. DISCUSSION AND CONCLUSION

The number of subdigraphs of a digraph representing a composite semantic Web service (based on our definitions) is equal to the cardinality of the power set $P(S)$ of its set of services. This is because we are considering every possible subset of the set of services.

The noticeable lack of a test collection of composite semantic Web services hinders the researchers to experiment their methods upon a valid collection of services. In this paper, we have proposed an adaptive approach of representing the digraphs of composite services using their adjacency matrix along with its related subdigraph calculation algorithm. We implemented our approach and evaluated its applicability on a public composite service as well as a synthetically created test set of composite services. Utilizing the proposed algorithm on an even small set of composite semantic Web services generates an enormous set of composite services.

We are currently finalizing the test collection generator's implementation. As a future work, we would like to deploy it for the public use of the research community.

ACKNOWLEDGMENT

This research is supported by University of Malaya High Impact Research Grant no. UM.C/625/HIR/MOHE/SC/13/2 from Ministry of Higher Education Malaysia and University of Malaya UMRG vote no. RP002F-13ICT.

REFERENCES

- [1] D. Booth, H. Haas, F. McCabe, E. Newcomer, M. Champion, C. Ferris, and D. Orchard. (February 2004). Web services architecture, technical representation, world wide web consortium. [Online]. Available: <http://www.w3.org/TR/ws-arch/>
- [2] S. A. McIlraith, T. C. Son, and H. Zeng. "Semantic web services," *IEEE Intelligent Systems*, vol. 16, no. 2, pp. 46-53, 2001.
- [3] SWS-TC. (2006). Semantic web services' test collection. [Online]. Available: <http://www.semwebcentral.org/projects/sws-tc/>
- [4] OWLS-TC. (2010). OWL-S service retrieval test collection. [Online]. Available: <http://www.semwebcentral.org/projects/owls-tc>
- [5] SAWSDL-TC. (2010). SAWSDL service retrieval test collection, [Online]. Available: <http://www.semwebcentral.org/projects/sawSDL-tc/>
- [6] L. Cabral, N. Li, and J. Kopecký, "Building the WSMO-Lite test collection on the SEALS platform," in *Proc. Second International Workshop on Evaluation of Semantic Technologies (IWEST)*, Heraklion, Greece, 2012, pp. 37-48.
- [7] D. Martin, M. Burstein, J. Hobbs *et al.* (2004). OWL-S: Semantic Markup for Web Services, Website, W3C Member Submission. [Online]. Available: <http://www.w3.org/Submission/OWL-S>
- [8] D. Martin, M. Burstein, D. McDermott *et al.*, "Bringing semantics to web services with OWL-S," *World Wide Web*, vol. 10, no. 3, pp. 243-277, 2007.
- [9] SAWSDL Working Group, *Semantic Annotations for WSDL and XML Schema*, 2007.
- [10] D. Fensel, F. Fischer, J. Kopecký, R. Krummenacher, D. Lambert, and T. Vitvar. (2010). WSMO-Lite: Lightweight semantic descriptions for services on the web. [Online]. Available: <http://www.w3.org/Submission/WSMO-Lite/>
- [11] U. Kuster and B. König-Ries, "Towards standard test collections for the empirical evaluation of semantic web service approaches," *International Journal of Semantic Computing*, vol. 2, no. 3, pp. 381-402, 2008.
- [12] H. Saboohi and S. A. Kareem, "A resemblance study of test collections for world-altering semantic web services," in *Proc. International Conference on Internet Computing and Web Services (ICICWS)*, Newswood Limited, Hong Kong, 2011, vol. 1, pp. 716-720.
- [13] K. J. Lin, J. Zhang, Y. Zhai, and B. Xu, "The design and implementation of service process reconfiguration with end-to-end

QoS constraints in SOA," *Service Oriented Computing and Applications (SOCA)*, vol. 4, no. 3, pp. 157-168, 2010.

- [14] H. Saboohi and S. A. Kareem, "Failure recovery of world-altering composite semantic services - a two phase approach," in *Proc. 14th International Conference on Information Integration and Web-based Applications & Services*, Bali, Indonesia, 2012, pp. 299-302.
- [15] H. Saboohi, A. Amini, T. Herawan, and S. A. Kareem, "Failure recovery of composite semantic services using expiration times," in *Proc. the First International Conference on Advanced Data and Information Engineering*, Malaysia, 2014, pp. 683-690.
- [16] H. Saboohi, A. Amini, and T. Herawan, "Increasing failure recovery probability of tourism-related webservices," *Recent Advances on Soft Computing and Data Mining Advances in Intelligent Systems and Computing*, vol. 287, pp. 517-527, 2014.
- [17] OPOSSum. (2009). Online portal for semantic services. [Online]. Available: <http://fusion.cs.uni-jena.de/OPOSSum/>
- [18] L. Cabral, N. Li, S. Tymaniuk, and D. Winkler, "D14.4 evaluation design and collection of test data for semantic web services tools v2," *Tech. Rep., Semantic Evaluation at Large Scale (SEALS)*, May 2011.
- [19] SEALS. (2011). SEALS - semantic evaluation at large scale. [Online]. Available: <http://www.seals-project.eu/>
- [20] J. Bang-Jensen and G. Z. Gutin, *Digraphs: Theory, Algorithms and Applications*, 2nd Edition, London: Springer, 2009.
- [21] H. Saboohi and S. A. Kareem, "An automatic subdigraph renovation plan for failure recovery of composite semantic web services," *Frontiers of Computer Science*, vol. 7, no. 6, pp. 894-913, 2013.



Hadi Saboohi received his B.Sc. and M.Sc. degrees in software engineering from Iran in 2001 and 2005 respectively, and a Ph.D. degree in computer science from University of Malaya in 2013. He is currently a postdoctoral research fellow at the Information System Department, Faculty of Computer Science and Information Technology, University of Malaya. He has published several papers in journals and conference proceedings and acts as a reviewer for various journals. His main research interests include Web intelligence, semantic Web services and data mining.



Amineh Amini received her B.Sc. degree in software engineering from Mashhad Azad University in 2001. She obtained her M.Sc. degree in the same field from Najafabad Azad University in 2005. She is a faculty member of Karaj Azad University from 2007. She received her Ph.D. degree from the Information System Department, Faculty of Computer Science and Information Technology, University of Malaya in 2014.

Her main research interests include data stream mining and big data and she has several conference and journal papers in her research area.



Tutut Herawan received the PhD degree in computer science in 2010 from Universiti Tun Hussein Onn Malaysia. He is currently a senior lecturer at the Department of Information System, University of Malaya. His research area includes rough and soft set theory, DMKDD, and decision support in information system. He has successfully co-supervised two PhD students and published more than 110 articles in various international journals and conference proceedings. He is an editorial board and act as a reviewer for various journals. He has also served as a program committee member and co-organizer for numerous international conferences/workshops.



Sameem Abdul Kareem received a B.Sc. degree in mathematics (Hons) from University of Malaya in 1986, a M.Sc. degree in computing from the University of Wales, Cardiff, UK in 1992, and a Ph.D. degree in computer science from the University of Malaya, Malaysia in 2002. She is currently an associate professor of the Department of Artificial Intelligence, Faculty of Computer Science and Information Technology, University of Malaya. Her research interests include medical informatics, machine learning, data mining and intelligent techniques. She has published over 100 articles in journals and conference proceedings.



Nooshin Anari received her B.Sc. degree in software engineering in 2012 from Pishva Azad University. Currently, she is a M.Sc. student in software engineering at Department of Computer, Kerman Islamic Azad University. Her main research interests include semantic web services and linked data.



Gholamreza Ahakian received his B.Sc. degree in hardware engineering from Azad University - Najafabad branch in Iran in 2001. He got the M.Sc. degree in computer architecture from science and research Azad University, Tehran, Iran in 2004. He is currently a faculty member of the Computer Department in Azad University Varamin - Pishva branch. His research focuses on computer architecture, multi value logic, VLSI designer.