GRD Journal for Engineering | Volume 2 | Issue 6 | May 2017
ISSN: 2455-5703

Improving Web Service Composition Technique using Graph Based Algorithm - A Technical Research

Urvi Chaudhary
Department of Computer Engineering
Ipcowala Institute of Engineering & Technology, Dharmaj

Mikin R Patel
Department of Computer Engineering
Ipcowala Institute of Engineering & Technology, Dharmaj

Rohan C Prajapati
Department of Computer Engineering
Ipcowala Institute of Engineering & Technology, Dharmaj

Abstract

Automatic web service composition is not a trivial problem, especially when the number of services is high and there are different control structures to handle the execution flow. Some approaches, treat the composition problem as a planning problem. In general, these approaches have important drawbacks such as high complexity, high computational cost and inability to maximize parallel execution of web services. Other approaches consider the problem as a search problem, where a search algorithm is applied over a graph or a tree in order to find a minimal composition. These proposals are simpler and more effective than the other approaches, and also many of them can exploit parallel execution of web services but it does not provide optimal result.
Keywords- Web Service, WSCC: p53a6307093, Ontology.xml

I. INTRODUCTION

Web services, adopted by Service Oriented Architecture (SOA), are loosely coupled reusable software components that semantically encapsulate discrete functionality and are distributed and programmatically accessible over the internet. It should facilitate integration of newly built applications both within and across organizational boundaries avoiding difficulties due to different platform, heterogeneous programming language. Web services can be used alone or in conjunction with other web services to carry out a complex aggregation or a business transaction. A web service is described using a standard that provides all of the details necessary to interact with the service such as, message formats, transport protocols, and location.

II. OBJECTIVES

- To study the importance of automatic web services composition process.
- To improve an algorithm for automatic web service composition.
- A new technique which utilize search method and AI planning method for getting best result.
- Compare the results of the new algorithm with the already existing basic algorithm and give the best results and optimal time.

III. PROBLEM FORMULATION AND ALGORITHM

Given a set of actions, their preconditions and positive and negative effects, a complete description of the initial state and a user goal find a sequence of actions achieving the goal.
For (P, A, I, G),
- P: set of propositions
- A: set of actions
- I : Initial State
- G: Goal propositions

IV. CHALLENGE.XML (A SET OF REQUESTS)

This contain the set of requests that is needed to the user. In that there is set of services that is defined as the composition routine. In composition routine here C2 set define in that there is no of services are provided and one resultant service are needed.

Table 1: Structure of Challenge.xml

WSChallenge		
CompositionRoutine		
Name	*Provided(10)*	*Resultant*
C2	*WSCC:p53a6307093, WSCC:p50a2654488, WSCC:p33a6894537*	*WSCC:p56a6099765*

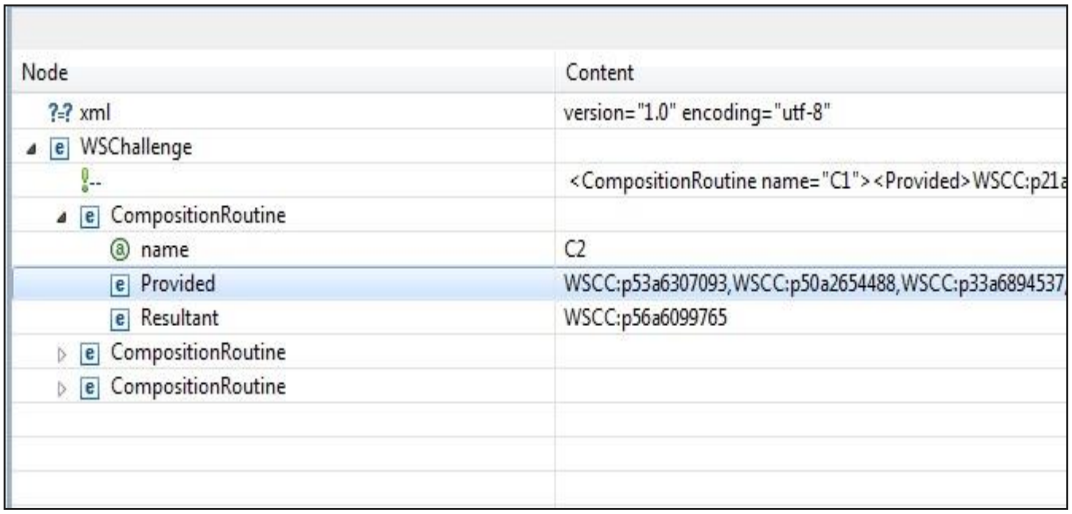

Fig. 1: Screen shot of Challenge.xml

A. *Ontology.xml (Taxonomy of Concepts)*

It provides the taxonomy of the web services. The taxonomy word defines the hierarchie of the web services that how they are connected with each other. It gives the brief view of how the service can composed through the connection of the various service nodes. In that whenever any instant are composed that are passed to these concepts and give the final services that respond user request.

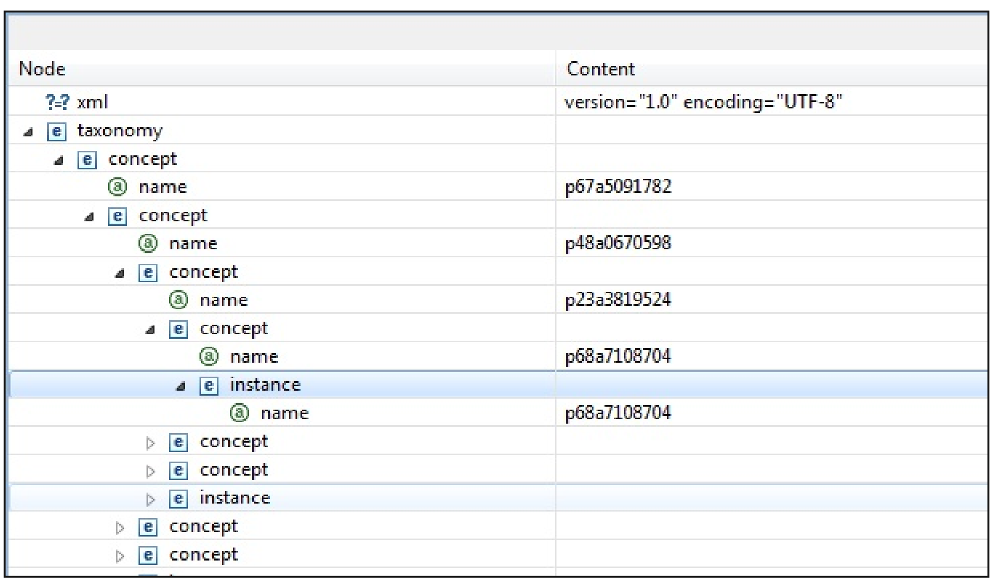

Fig. 2: Screen shot of Ontolgy.xml

V. RESULT ANALYSIS

A. Algorithm Analysis

Let (O; s0; g) be a planning problem, where O is a set of operators (each action is a ground operator), s0 is the initial state, and g is a set of goal propositions. Suppose that the planning problem has n propositions and m actions. It follows that the number of elements that are contained in every proposition set and action set are no more than n and m .Every simplified planning graph has a fixed point level k, reason for this is that its proposition sets and action sets monotonically increase from one level to the next, while the number of elements that are contained in every proposition set and action set is bounded; thus, every simplified planning graph must reach a fixed point level. Therefore, the complexity of the simplified planning graph is polynomial, because it has finite distinct levels and the number of elements that are contained in every proposition set and action set is no more than n and m.

Table 2: Experiment Results with Comparison

Initializing Time	*832*
Concepts size	*1578*
Things size	*3102*
Parameter size	*6309*
Services size	*572*

B. Result of Composition Request after Pruning

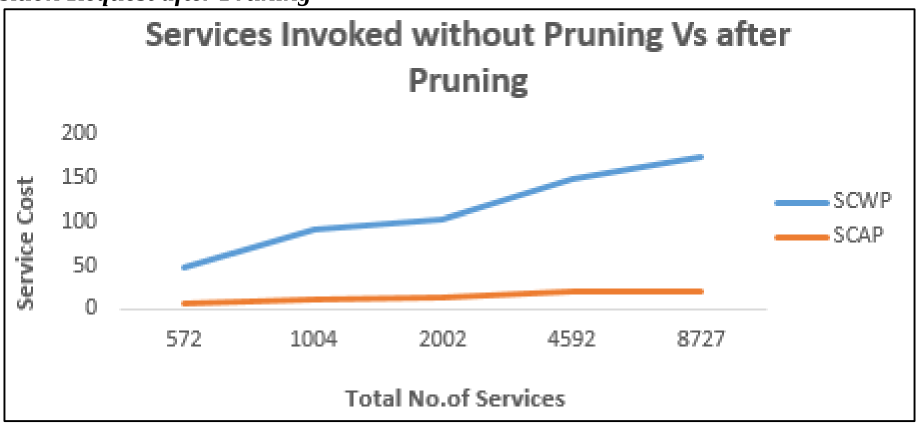

Fig. 3: Services Invoked Without Pruning Vs after Pruning

As we are observing in graph we require more service if we are not utilizing pruning techniques because of redundant services are invoked so we have used pruning technique to remove redundant services.

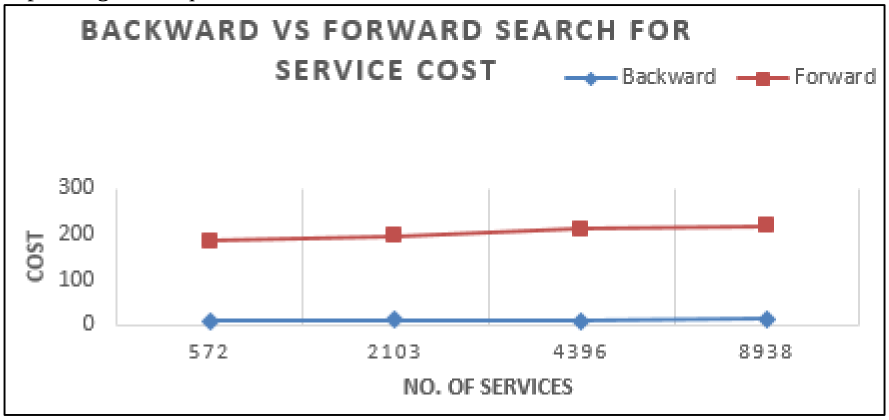

Fig. 4: Backward Vs Forward Search for Service Cost

As shown in above Figures the proposed algorithm has much better usage of Web services in all cases in terms of obtaining the solution. Because of the aim of the forward algorithm, which is to reach the goal as quick as possible, it expands the search space in planning graph for services composition problem no matter how many redundant Web services are produced.so service cost will be much higher than backward search.

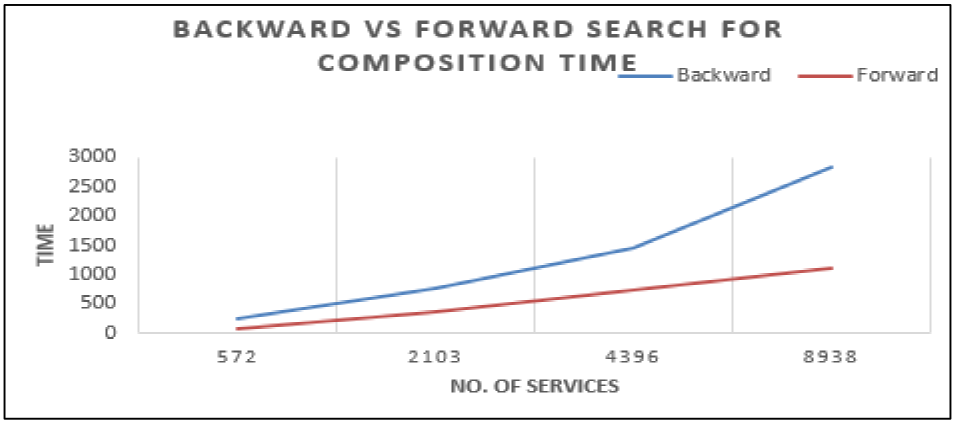

Fig. 5: Backward Vs Forward Search for Composition Time

As shown in above figure backward search algorithm needed more time to compose a service because of it will search back from goal state to initial state.

VI. Conclusion

We have present an efficient Semantic Web service composition algorithm based on a simplified planning graph. For a solvable problem, it can find a solution in polynomial time, but with possible redundant Web services. We apply four strategies to make the solution concise. We also compare result with backward and forward search algorithm. This approach opens a new path to improve the planning graph when we solve the Web service composition problem.

References

[1] Mansour H. Elfawal and T. Dillon, "Dependability and Rollback Recovery for Composite Web Services," IEEE TRANSACTIONS ON SERVICES COMPUTING, vol. 4, no. 4, pp. 328 - 339, OCTOBER-DECEMBER 2011.
[2] J. J. Hu, X. Zhao, Y. D. Cao, Research on Transaction Web Service Selection Algorithm inWSC, Applied Mathematics & Information Sciences, 7, 725-731 (2013).
[3] Rajesh Karunamurthy, Ferhat Khendek, and Roch H. Glitho, "A novel architecture for Web service composition," Journal of Network and Computer Applications , vol. 35, no. 2, pp. 787–802, March 2012.
[4] YU Qing-mei, WANG Lan, and HUANG Dong-mei, "Fishery Web Service Composition Method Based on Ontology," Journal of Integrative Agriculture, vol. 11, no. 5, pp. 792–799, May 2012.
[5] Rui Wang et al., "Web Service Composition using Service Suggestions," in 2011 IEEE World Congress on Services, Washington, DC, 2011, pp. 482 - 489.
[6] D. Paulraj, S. Swamynathan, M. Madhaiyan, Process Model-based Atomic Service Discovery and Composition of Composite Semantic Web Services using Web Ontology Language for Services (OWL-S), Enterprise Information Systems, 6, 445-471 (2012).
[7] Elham Paikari, Emadoddin Livani, Mohammad Moshirpour, Behrouz H Far, and Günther Ruhe, "Multi-Agent System for Semantic Web Service Composition," in Knowledge Science, Engineering and Management: Springer Berlin Heidelberg, 2011, pp. 305-317.
[8] Shanfeng Qi, Xinhuai Tang, and Delai Chen, "An Automated Web Services Composition System Based on Service Classification and AI Planning," in 2012 Second International Conference on Cloud and Green Computing, Xiangtan, 2012, pp. 537 – 540.
[9] Guisheng Fan, Huiqun Yu, Liqiong Chen, and Dongmei Liu, "Petri net based techniques for constructing reliable service composition," The Journal of Systems and Software, vol. 86, no. 4, pp. 1089–1106, April 2013.
[10] Furkh Zeshan and Radziah Mohamad, "Semantic Web Service Composition Approaches: Overview and Limitations," International Journal on New Computer Architectures and Their Applications, pp. 640-651, 2011.
[11] P. Rodriguez-Mier, A. Gonzalez-Sieira, M. Mucientes, M. Lama, and A. Bugarin, "Hipster: An Open Source Java Library for Heuristic Search," in 9th Iberian Conf. on Information Systems and Technologies, 2014.
[12] P. Rodriguez-Mier, M. Mucientes, J. Vidal, and M. Lama, "An Optimal and Complete Algorithm for Automatic Web Service Composition," Int. J. of Web Services Research (IJWSR), vol. 9, no. 2, pp. 1–20, 2012.
[13] D. Lee, J. Kwon, S. J. Lee, S. Park and B. Hong, "Scalable and Efficient Web Services Composition Based on a Relational Database", Journal of Systems and Software, vol. 84, (2011), pp. 2139-2155.
[14] X. Z. Wang, Z. J. Wang, X. F. Xu and Y. Liu, "A Service Composition Method for Tradeoff Between Satisfactions of Multiple Requirements", Journal of Computer Research and Development, vol. 48, no. 4, (2011), pp. 627-637.
[15] W. L. Ding, Q. Wang and S. Zhao, "A User-Centric Service Composition Method Synthesizing Multiple Views", Chinese Journal of Computers, vol. 34, no. 1, (2011), pp. 131-142.