



A Survey & Analysis of Web Service Composition

S. Satpathy¹, A. K. Tripathy^{1,2}, S. K. Pradhan² and M. R. Patra³

¹Dept. of CSE, Silicon Institute of Technology, Odisha, INDIA

²Dept. of Computer Science & Application, Utkal University, Odisha, INDIA

³Dept. of Computer Science & Application, Berhampur University, Odisha, INDIA

Abstract: Web Services have become a promising technology to design & build complex enterprise business applications out of single web based software components. Existing technologies of Web Services are extended to give value added customised services to users through composition, which are developed by different organisations & offer diverse functionalities, behavioural properties & Quality of Service (QoS) values. Web Service Composition is a challenging research issue which is a process to construct a composite Web Service recursively as a workflow of other existing Web Services. The objective of this article is to overview the issues of Web Service Composition & analysis of existing approaches.

Keywords: Web Services; SOA; Quality of Services; Dynamic Web Service Composition

I. INTRODUCTION

A Web Service technology is a method of communication between two software modules over the World Wide Web which defines the Web Service as a software system designed to support inter-operable machine-to-machine interaction over a network. Service Oriented Architecture (SOA) is a set of principles for designing & developing software in the form of Web Services. The Web Service is the implementation of SOA allowing the construction & sharing of independent software. It is a software system identified by a URL, whose public interfaces and bindings are defined and described using XML [3].

Research on SOA brings a promising technique to create applications composed by dynamically selected individual Web Services. This technique is known as Web Service Composition. Web Services involved in a composition process are designed & handled by independent third parties. Therefore, composition assumptions made at design time may violate at run time. With the growing nature of Web Service providers & dynamic nature of Service Based Systems, the dynamic service composition is needed. With the demand of dynamic service composition, appropriate service selection among many available services becomes vital. This process is called service selection for composition. In order to find a distinctive edge over other competitors, automatic dynamic service composition can be applied to further adapt to the requirements of the users.

Many researchers had been focused on different aspects of Web Service Composition. Automatic Dynamic Web Service composition research categorized to Automatic composition [38][24][20], Dynamic Web Service composition [9][12][21], transactional Web Service composition [2][16][24], QoS-aware composition [12][17][10][23] & user constraints and performance oriented Web Service composition [7][8][29].

The objective of this article is to overview Web Services & Web Service composition. Specifically, the main focus is given to QoS-aware service composition & service clustering. Further, we highlight the existing research issues in different approaches.

In Section II, we discuss different research issues in Web Service composition. Section III describes various composition approaches. The paper shows the survey on Dynamic Web Service Composition in Section IV and various clustering approaches in Section V. And finally Section VI concludes this analysis paper.

II. DIFFERENT RESEARCH ISSUES IN WEB SERVICE COMPOSITION

With the steadily growing number of service providers the composition becomes more & more intense. In order to find a distinctive edge over other competitors, service composition can be applied to further adapt to the requirements of the users. Before performing Web Service composition, some basics have to be performed. There are certain issues that have a large impact on service composition. The following descriptions do not claim completeness of the issues. The most important composition issues are listed below.

A. Composition Automation

In today's Web, Web Services are created and updated on the fly. It's already beyond the human ability to analyze them and generate the composition plan manually. So the manual service composition is hectic & error prone task. To make SOA available & acceptable globally, automatic service composition is essential.

B. Workflow Management

To achieve complex workflow applications by composition, a simple workflow management environment for service composition is required which may help to achieve current & robust functional properties.

C. Functional & Non-functional (QoS) constraints

The dynamic & loosely coupled nature of SOA makes it more popular where as the functional constraints assumed to achieve before composition may violate at run time due to fault in component services. Therefore to achieve consistent functional properties of SBS is a real challenge. Designing a composite Web Service to ensure the optimal quality of services (QoS) also remains an important challenge. Web Service composition based on functional properties ensures a reliable execution. But to reduce the set of possible transactional and functional issues, QoS aware service selection is mandatory though it refers the clients' requirements.

D. SBS monitoring & run-time adaptations

The monitoring phase is one of the most important parts of the Web Service lifecycle, as it enables gaining a clear view of how services perform within their operational environments. If any error occurs after the SBS monitoring, which cannot be modified, then the run time adaptation issue is needed which may adapt a new appropriate Web Service by replacing the error prone one.

E. SLA specification & monitoring

Service Level Agreement (SLA) is defined upon a business process as its end-to-end QoS constraints define how abstract services interact to accomplish a certain business goal. To achieve an error free environment, service monitoring requires that a set of QoS metrics be gathered on the basis of SLA.

III. COMPOSITION APPROACHES

A. Static Service Composition

Static composition takes place during design-time when the architecture and the design of the software system are planned. The components to be used are chosen, linked together, and finally compiled and deployed. This may work fine as long as the Web Service environment & service component does not, or only rarely, change. If other businesses provide newer services, or the old services are replaced by other ones, inconsistencies might be caused. Then it is unavoidable to change the software architecture or, in the worst case, even change the process definition & redesign the system. In this case, static composition may be too restrictive & components should automatically adapt to unpredictable changes.

B. Dynamic Service Composition

The service environment is a highly flexible and dynamic environment. New services become available on a daily basis and the number of service providers is constantly growing. Ideally, service processes should be able to transparently adapt to environment changes, and to customer requirements with minimal user intervention.

C. Automatic Service Composition

With the steadily growing number of service providers the composition becomes more & more intense. In order to find a distinctive edge over other competitors, automatic service composition can be applied to further adapt to the requirements of the users. This approach aims at automatically composing services to satisfy the requests of users according to the information of users input.

D. Dataflow driven Service Composition

Data-flow requirements are an important aspect of service composition. Although several approaches have been proposed to specify data-flow requirements, they cannot be efficiently exploited in dynamic setting [11]. The Model Driven Service Composition [26] facilitates the management and development of dynamic service compositions. UML (Unified Modeling Language) is used to provide a high level of abstraction, and to enable direct mapping to other standards, such as BPEL4WS. The OCL (Object Constraint Language) is used to express business rules and to describe the process flow.

IV. DYNAMIC WEB SERVICE COMPOSITION

Dynamic Web Service Composition has been proposed by Hwang, et. al [9]. They have formulated the dynamic Web Service selection problem in a dynamic and failure-prone environment. They proposed to use a Finite State Model (FSM) to model the invocation order of operations in each Web Service and to construct a Web Service composition that enumerates all possible delegations. They defined a measure, called AR, to determine the probability that the execution of operations starting from a configuration will successfully terminate. Two Web Service selection strategies, AR-based and CAR-based, were proposed. The experimental results showed that the AR-based strategy tops in success rate, while the CAR based strategy achieves the best compose ability rate at the cost of a slightly poorer success rate. But the proposed solution is based on an orchestration model, rather than a choreography model. It assumes each operation has an equal probability of being chosen and only reliability is considered in this paper.

Manikrao, et. al. [21] proposes a dynamic Web Service selection framework which combines a recommendation system with semantic matching of service requirements. The matching service can be a single service or composition of registered services. The recommendation system is based on user feedback and collaborative filtering techniques. It helps the user in selecting a web service from a set of similar services. The

Model Driven Architecture (MDA) approach is useful to ease code generation for web service providers and to specify user requirements.

A. TQoS aware service composition

While many works have been done for Web Service selection, designing a composite Web Service to ensure not only correct and reliable execution but also optimal QoS remains an important challenge [23]. Indeed, Web Services composition based on transactional properties ensures a reliable execution, however, an optimal QoS composite Web service is not guaranteed. Moreover, composing optimal QoS Web services does not guarantee a reliable execution of the resulting composite Web Service. Thus, QoS-aware and transactional-aware should be integrated. However, the problem is generally addressed from the QoS side or from the transactional side separately. The conventional QoS aware composition approaches [23] [10][33][38] do not consider the transactional constraints during the composition process, likewise transactional-aware ones[2][16][24] do not consider QoS.

Liu, et. al.[17] proposed a composition model in design time, which captures both aspects in order to evaluate the QoS of a composite WS with various transactional requirements. However, they do not consider the automatic selection step and only analyze the impact of the transactional requirements on the QoS of the composite WS. Hadded, et. al.[7] have proposed a composition approach that is TQoS-driven approach. It consists of a Web Service selection approach which is realized depending on transactional and QoS user requirements. The former is established by means of a risk notion that indicates if the results can be compensated or not. The latter is expressed as a weight over each QoS criterion. They proposed and analyzed a selection algorithm based on the workflow patterns and the transactional properties of the component web services. They have evaluated the scalability of the TQoS algorithm. The experimental results show that the number of activities does not affect the computation cost of the algorithm since the selection is done incrementally. But more experiments are needed to consider different scenarios and compare the performance of their algorithm with related ones.

B. Service composition with service dependent QoS attributes

Feng, et. al.[8] proposed a composition approach extending the dependency of QoS attributes. They considered the QoS aware service composition problem in the presence of service dependent QoS. By formally modeling service-dependent QoS that captures both partial and full dependencies, they proposed a method that dynamically refines the composed workflow in light of QoS dependencies and user-provided topological and QoS constraints. They argued that the method effectively handles service-dependent QoS by directly integrating it into the composition process, rather than after the composition. In addition, the method supports topological and aggregated QoS constraints to enable user-guided composition. Their approach is capable of offering significant improvements in performance on real life scenarios with complex service and QoS dependencies.

C. Service composition with clustering approaches

Automatic Web Services composition approaches aim at automatically composing services to satisfy users' request according to the information of users' input. As the number of Web Services is increasing in an exponential way, the performance of automatic service composition is faced with challenges. It is not possible to be scalable and to check each single service with its QoS features whether it is appropriate for the Web Service Selection or not. So to make the scenario scalable, Clustering approach is used. The approach firstly clusters all the Web Services registered on semantic parameter level, and then finds the most suitable service in each cluster to keep the top Web Services which are not dominated by other services.

V. CLUSTERING BASED DYNAMIC WEB SERVICE COMPOSITION

A cluster is a group of similar objects and it is not similar to other clusters. The goal of clustering is a quick access to the web services. The common trend to achieve the automation is to annotate the services with additional semantic information which is called as semantic annotation, and mix the information retrieval computation with logic based reasoning to cluster the similar services[13][14][5]. The automation can be achieved in Web Service clustering & in QoS-based selection for Web Service composition [37][35]. While much work has been done on clustering for Web Service discovery and matching, little research has considered clustering for Web Service composition according to the compose ability.

The approaches [31][29][20][36] propose different kinds of solutions for QoS oriented service selection for dynamic composition. A multi-grain clustering and selection model for service composition [36] considers the requirement of customers in service composition in the end-user view. The paper gives a formal specification on model using the methodology deriving from granular computing. This model is more understandable for an end-user and conforms to the intuitive granular cognition mode. A methodology from granular computing has been introduced [36] to theoretically construct the model. The basic ideas of granular computing-problem solving with different granularities have been explored in many literatures and papers [34] in many fields, such as artificial intelligence, interval analysis, rough set theory, and cluster analysis, etc. But in [19] the information processing methodology of granular computing has been used to build the service composition clustering and selection model.

Clustered web service composition has been proposed by Chen et al[20], who proposed an algorithm based on backward chaining method to implement automatic service composition. They cluster services based on their output parameters to improve the performance of the algorithm. In each cluster, they sort services and find skyline services to improve QoS of results. To avoid the waste of memory, they have also adapted the backward chaining method to an enhanced version that services chain is found recursively. In the stage of service composition, they merge the clusters into it and search matching services cluster by cluster. Experiments have shown that the performance of Clustering Based Backward Chaining Method outperformed Backward Chaining Method and composition method through selecting hybrid Skyline Services in most scenes.

Wagner, et. al.[31] proposed a data structure which arranges functionally similar services in clusters and computes the QoS of each cluster. This way, the utility in general and the reliability of the composed workflows are significantly increased. They proved the feasibility and the benefits of functional clustering. They considered the issues like enhanced reliability, QoS-awareness, compliance with QoS constraints & flexibility. They first identified clusters with functionally similar services with the belief that in a realistic setting, services are often created with the same intended purpose, having yet slightly different interfaces. By identifying these clusters they pruned the search space and moreover determine which services have most backup services.

Several approaches have been proposed that incorporate QoS awareness with service planning. Chen et. al. [20] integrates a Markov decision process model into the HTN planer SHOP2. The decision tree is pruned during the decomposition phase if the QoS of a branch is below a threshold. This way the utility is improved, but QoS constraints and backup services are not considered in this work.

Lecue, et. al. [15] presents the architecture of a system that generates workflow templates from previous execution logs by using process mining techniques. The approach optimizes the QoS utility of the workflow and takes constraints into account. However, execution logs are required in this approach and QoS aspects are not incorporated in the template generation phase. Therefore, services with low utility might be selected, compromising the utility of the computed workflow. That is why they presented an approach that combines a functional pre-clustering of services with automatic service composition. The clusters in the computed graph indicate possible backup services for each registered service. The evaluations show that the reliability and the utility of the computed workflows are significantly increased. They discussed how service selection algorithms can be applied in order to achieve QoS constraint awareness. But in their approach the clusters are of small sizes. The heuristic methods should be analyzed to increase the size of the clusters.

As per our analysis the automatic Web Services composition approaches aim at automatically composing services to satisfy users' request according to the information of users' input. As the number of web services is increasing in an exponential way, the performance of automatic service composition is faced with challenges. It is not possible to be scalable and to check each single service with its QoS features whether it is appropriate for the Web Service Selection or not. So to make the service selection process scalable, clustering approaches are useful. In many compositional scenarios with client dynamic QoS oriented service selection process; there are chances of ignorance of selection of optimal combination of component services due to incompatibility between them. In these cases, introduction of third connector service to make them compatible may produce a global optimal selection.

VI. CONCLUSION

The research on Web Service Composition has started a quite long time ago and numerous problems have been solved already. The composition strategies built till now are far from being perfect. These are still at a very early stage of development. In this article we overviewed the issues of dynamic Web Service Composition & analysis of existing approaches. We present various kinds of composition strategies, QoS aware composition techniques, clustering approaches, their benefits and have given an outlook to essential future research work. We conclude the article by mentioning some current research issues in dynamic service composition to make it preferable for global optimization.

REFERENCES

- [1] G. Alonso, F. Casati, H. Kuno, and V. Machiraju. Web services: Concepts, architectures and applications. 2004.
- [2] S. Bhiri, O. Perrin, C. Godart, et al. Extending workflow patterns with transactional dependencies to define reliable composite web services. In *Advanced International Conference on Telecommunications and International Conference on Internet and Web Applications and Services (AICT/ICIW 2006)*, pages 145–150, 2006.
- [3] D. Booth, H. Haas, F. McCabe, E. Newcomer, M. Champion, C. Ferris, and D. Orchard. Web services architecture, w3c working group note, 11 february 2004. *World Wide Web Consortium*, article available from: <http://www.w3.org/TR/ws-arch>, 2004.
- [4] M. A' Corella and P. Castells. A heuristic approach to semantic web services classification. In *Knowledge-Based Intelligent Information and Engineering Systems*, pages 598–605. Springer, 2006.
- [5] M. A' Corella and P. Castells. Semantic-based taxonomic categorization of web services. In *1 st International Workshop on Semantic Matchmaking and Resource Retrieval*, page 105, 2006.
- [6] S. Dustdar and W. Schreiner. A survey on web services composition. *International journal of web and grid services*, 1(1):1–30, 2005.

- [7] J. El Hadad, M. Manouvrier, and M. Rukoz. Tqos: Transactional and qos-aware selection algorithm for automatic web service composition. *Services Computing, IEEE Transactions on*, 3(1):73–85, 2010.
- [8] Y. Feng, L. D. Ngan, and R. Kanagasabai. Dynamic service composition with service-dependent qos attributes. In *Web Services (ICWS), 2013 IEEE 20th International Conference on*, pages 10–17. IEEE, 2013.
- [9] S.-Y. Hwang, E.-P. Lim, C.-H. Lee, and C.-H. Chen. Dynamic web service selection for reliable web service composition. *Services Computing, IEEE Transactions on*, 1(2):104–116, 2008.
- [10] M. C. Jaeger, G. Rojec-Goldmann, and G. Muhl. Qos aggregation for web service composition using workflow patterns. In *Enterprise distributed object computing conference, 2004. EDOC 2004. Proceedings. Eighth IEEE International*, pages 149–159. IEEE, 2004.
- [11] R. Kazhamiakin, A. Marconi, M. Pistore, and H. Raik. Data-flow requirements for dynamic service composition. In *Web Services (ICWS), 2013 IEEE 20th International Conference on*, pages 243–250. IEEE, 2013.
- [12] F. H. Khan, S. Bashir, M. Y. Javed, A. Khan, and M. S. H. Khiyal. Qos based dynamic web services composition & execution. *arXiv preprint arXiv:1003.1502*, 2010.
- [13] N. Kokash, W.-J. van den Heuvel, and V. DAndrea. Leveraging web services discovery with customizable hybrid matching. In *Service- Oriented Computing-ICSOC 2006*, pages 522–528. Springer, 2006.
- [14] R. Lara, M. A. Corella, and P. Castells. A flexible model for web service discovery. In *1 st International Workshop on Semantic Matchmaking and Resource Retrieval*, page 51, 2006.
- [15] F. L’ecu’e, Y. Gorronogioita, R. Gonzalez, M. Radzinski, and M. Villa. Soa4all: An innovative integrated approach to services composition. In *Web Services (ICWS), 2010 IEEE International Conference on*, pages 58–67. IEEE, 2010.
- [16] L. Li, C. Liu, and J. Wang. Deriving transactional properties of compositeweb services. In *Web Services, 2007. ICWS 2007. IEEE International Conference on*, pages 631–638. IEEE, 2007.
- [17] A. Liu, L. Huang, and Q. Li. Qos-aware web services composition using transactional composition operator. In *Advances in Web-Age Information Management*, pages 217–228. Springer, 2006.
- [18] A. Liu, Q. Li, L. Huang, and M. Xiao. Facts: A framework for faulttolerant composition of transactional web services. *Services Computing, IEEE Transactions on*, 3(1):46–59, 2010.
- [19] Y.-s. Luo, Y. Qi, L.-f. Shen, D. Hou, C. Sapa, and Y. Chen. An improved heuristic for qos-aware service composition framework. In *High Performance Computing and Communications, 2008. HPCC’08. 10th IEEE International Conference on*, pages 360–367. IEEE, 2008.
- [20] Y. Ma, L. Chen, J. Hui, and J. Wu. Cbbcm: Clustering based automatic service composition. In *Services Computing (SCC), 2011 IEEE International Conference on*, pages 354–361. IEEE, 2011.
- [21] U. S. Manikrao and T. Prabhakar. Dynamic selection of web services with recommendation system. In *Next Generation Web Services Practices, 2005. NWeSP 2005. International Conference on*, pages 5–pp. IEEE, 2005.
- [22] B. Medjahed and A. Bouguettaya. A multilevel composability model for semantic web services. *Knowledge and Data Engineering, IEEE Transactions on*, 17(7):954–968, 2005.
- [23] D. Menasce. Composing web services: A qos view. *Internet Computing, IEEE*, 8(6):88–90, 2004.
- [24] F. Montagut, R. Molva, and S. T. Golega. Automating the composition of transactional web services. *International Journal of Web Services Research (IJWSR)*, 5(1):24–41, 2008.
- [25] R. Orfali, D. Harkey, and J. Edwards. *Client/server survival guide*. John Wiley & Sons, 2007.
- [26] B. Orri’ens, J. Yang, and M. P. Papazoglou. Model driven service composition. In *Service-Oriented Computing-ICSOC 2003*, pages 75–90. Springer, 2003.
- [27] J. C. Schlimmer et al. Web services description requirements. *Web Service Description Working Group report*, <http://www.w3.org/TR/wsdesc- reqs>, 2002.
- [28] D. Skoutas, D. Sacharidis, A. Simitsis, and T. Sellis. Ranking and clustering web services using multicriteria dominance relationships. *Services Computing, IEEE Transactions on*, 3(3):163–177, 2010.
- [29] A. K. Tripathy and M. R. Patra. Cost effective, requirement oriented, web service composition and adaptation. *International Journal on Recent Trends in Engineering & Technology*, 5(1), 2011.
- [30] A. K. Tripathy and M. R. Patra. Service based system monitoring framework. *International Journal of Computer Information Systems and Industrial Management Applications: IJCISIM*, 3:924–931, 2011.
- [31] F. Wagner, F. Ishikawa, and S. Honiden. Qos-aware automatic service composition by applying functional clustering. In *Web Services (ICWS), 2011 IEEE International Conference on*, pages 89–96. IEEE, 2011.
- [33] B. Wu, C.-H. Chi, and S. Xu. Service selection model based on qos reference vector. In *Services, 2007 IEEE Congress on*, pages 270–277. IEEE, 2007.
- [34] Y. Yao. The art of granular computing. In *Rough Sets and Intelligent Systems Paradigms*, pages 101–112. Springer, 2007.
- [35] T. Yu, Y. Zhang, and K.-J. Lin. Efficient algorithms for web services selection with end-to-end qos constraints. *ACM Transactions on the Web (TWEB)*, 1(1):6, 2007.
- [36] L. Yuan-sheng, Q. Yong, H. Di, C. Ying, and S. Lin-feng. A clustering and selection model for service composition using granular computing. In *Industrial Electronics and Applications, 2009. ICIEA 2009. 4th IEEE Conference on*, pages 3077–3082. IEEE, 2009.
- [37] L. Zeng, B. Benatallah, A. H. Ngu, M. Dumas, J. Kalagnanam, and H. Chang. Qos-aware middleware for web services composition. *Software Engineering, IEEE Transactions on*, 30(5):311–327, 2004.