Adaptive QoS-Aware Web Service Composition

Deivamani Mallayya and Baskaran Ramachandran

Anna University, Chennai, 600 025, India
{deivamani,baaski}@cs.annauniv.edu

Abstract. Service oriented architecture is a challenging area to fervently focus on. In that web service composition plays a vital role. The main crux behind composition lies on the effective selection of available web services in order to provide the value added services on the fly. Quality of Service (QoS) is one of the non functional properties of the web services, which is used to evaluate the degree to which the service can satisfy the service request. In the proposed approach, the composition is handled based on the QoS the web service has provided in its previous attempts towards composition. A separate process of updating the beliefs and reputation is been identified which stores the appropriate belief factor against the candidate web service in the process registry. Instead of having the QoS as a constant provider specified value, our approach assigns the value based on the end users feedback. The paper discusses the approach used in identifying the quality of the web service composition and the efficiency of composing relevant services for the service request.

Keywords: Service Composition, Quality-of-Service(QoS), Belief, Graph Construction, Service Oriented Architecture.

1 Introduction

Web services are software application intended to do a particular functionality and thereby obeys all the open standards of service oriented architecture namely WSDL (Web Service Description language) for description of the web service, UDDI (Universal Description, Discovery and Integration) to register the web service to the registry and SOAP (Simple Object Access Protocol) to enquire and access the web services. The service provider creates and registers the web service with the registry. When there is a request for a service, the registry is searched to find the appropriate web service and the same is been delivered back to the requester [1].

Composition was innovated as a single web service could not assure the entire need of the requester. Composition is a service designed from a set of selected web services. Initially, composition was introduced to please the novice's but now it has been welcomed by people of all knowledge level. The main crux of the composition lies behind the effective selection of the web services. Once the services have been identified, the flow between those services has to be judged. This flow of information from one web service to the other should happen synchronously.

N. Meghanathan et al. (Eds.): CCSIT 2012, Part II, LNICST 85, pp. 488–497, 2012.

Based on the time and methodology by which the candidate web services are transferred as composition, they are named as manual, semi-automatic, automatic or dynamic. The proposed approach concentrates on dynamic composition. Some of the requirements for dynamic web service composition includes: automation, dynamicity, semantic capabilities, QoS awareness, partial observing ability, correctness, scalability, domain independence, and so on. The degree to which these requirements are incorporated determines the efficiency of composition.

Automation is innovated with the intention to minimize the user intervention and thereby minimize the time spent, eliminate the human errors which in turn reduces the overall cost of the system. Dynamicity is the ability of the system to provide a composite service at run time. Compared to static composition, dynamic service composition overcomes the issues like binding the web service which is no longer available at run time (which were available at design time), the services being replaced by the new ones (once they are published by the providers).

Several approaches have been analyzed to provide composition. Semantic capabilities with web services are the much targeted area in SOA [2]. Compared to syntactic similarity, composition based on the semantic similarity gives much meaningful composition. Various tools have been devised to construct ontology, from which the semantics of the service request is been analyzed and selection of candidate web services is been done from the registry. Apart from the core functionality of the web services, the non functional properties like Quality of service (QoS) are also taken into account. The importance of having QoS properties at the constraint in selecting a candidate web service occurs, when the functional requirement stated by the service request is satisfied by more than one service. So if QoS properties are used to compose the web services, we could ensure the composition not only guarantee the service been requested, but also promises quality in the same.

2 Related Work

The QoS parameters of the web service are used in service composition in many ways. The candidate web services are composed to form a composition by plug and play technology. J.L. Pastrana et al has presented a tool for composition which connects the other candidate web services with connectors to which the web services are plugged in at the time of composition [3]. The adaption of the web services to the connectors was done by using OWL ontology.

Ping Wang et al proposed a new method for composition by selecting the correct set of candidate service based on QoS done based on fuzzy linear programming which helps in identifying dissimilarity of service alternatives, selecting the correct candidate service with respect to the users preference and acceptance level [4][5].

Once the services are identified, ranking among them is the most vital task. Ranking among the web services are handled with QoS, a measure of requester’s satisfaction or experience with the service [6][11]. Ranking of web service has to be done in the view point of users and system brokers. The relation between the different QoS properties also has to be taken to account. LIU Feng has proposed his research on user aware QoS based web service composition by using quantum based genetic algorithm[9]. Yuan-sheng Luo et al demands that QoS properties which are been locally used for selecting the web service alone would not be sufficient for an efficient web service [15].

3 Service Composition Architecture

A sole web service by itself could not guarantee the intend expectation of the user. And hence the innovation of Web service composition came to picture. **Composition** is a service coined from several candidate web service, served to the requester as a response to the service request. This composition could be carried out in many ways, by using semantics, context, aspects and many more. The main intension behind providing the service composition is to serve the services to the requester in a more efficient manner, and thereby promote the trust and reputation of the service provider with respect to each candidate web service. The driving wheel behind composition is on timing and synchronization of web services.

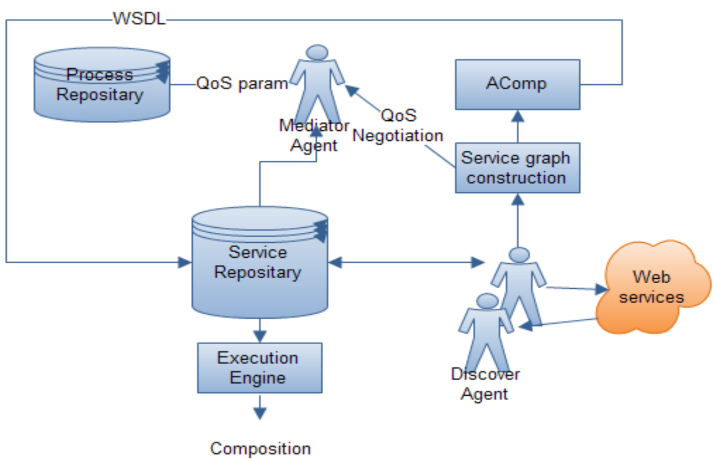

Fig. 1. Architecture of Adaptive QoS- aware Web Service Composition

Our approach to web service composition is based on the non functional property, the QoS. Fig 1 sketches the overall architecture of Adaptive QoS- Aware Web Service Composition. The system consists of five subcomponents: the process repository, the service repository, Service Graph Construction, A-Comp and Execution Engine.

Quality of service properties are used to select the candidate web services from the registry. Based on the type of application, the system is going to function in; the selection of QoS might get changed. In common some of the QoS properties to mention include: availability, importance, cost, runtime, success ratio, trustworthiness, security and so on.

3.1 Construction of QoS-Domain Ontology

The selection of candidate web service and the composition is done with the help of ontology. Ontology is a representation of concepts, their properties and relationship among the concepts. The proposed approach uses extended domain ontology with QoS properties included. A sample rdf format is been shown below [8][14].

```
<webService rdf:id="trainAvailability">
  <functional rdf:id="class">
     ………….
  </functional>
  <NonFunctional rdf:id="QoS">
    <QoSParam rdf:id="Availability"/>
      <hasImpact>
        <QoSImpact rdf:id ="VeryHigh" />
      </hasImpact>
      <hasRelationship>
        <relationship rdf:id="performance/>
      </hasRelationship>
      <hasMetric>
        <metric rdf:id="successRate"/>
      </hasMetric>
    </QoSParam>
  </NonFunctional>
</webService>
```

3.2 Composition of Web Services

The composition of web services is performed by three steps namely: Selection of web services, Graph Construction and Best path Identification.

Selection of Web Services. A web service composition WS_c, is a service coined from a subset of web services registered in the registry which satisfies the QoS requirements stated by the user. The requirements are specified in terms of availability, cost, success rate, trust, security and performance. The selection of QoS parameters favors both the provider and the beneficent [7]. The level of QoS parameter that is being expected by the requester has to be determined from the user request. The selection of web services is handled by the selection agent with the level of QoS needed as the input. In our proposed approach of service selection, all the services which satisfy the determined QoS level are been selected.

We define a Web Service Composition problem A-Comp as

- Let N = $\{WS_1, WS_2, WS_3, \ldots WSn\}$ be the set of web services registered in the registry.
- Web service WS = (WS_functional_params, WS_Non_functional_params)
- DeterminedQoS = ($Service_{Req}$, $Expected_{Res}$)
- WS_{Select} = select(WS_Non_functional_params, DeterminedQoS)

Based on the service request $Service_{Req}$ and the expected response $Expected_{Res}$ from the requester, the DeterminedQoS is been identified. Considering a web service as a set of functional and non functional parameters, web services, which has a value of non functional parameter greater than the DeterminedQoS are selected.

Graph Construction

As shown in the Fig 2, graph is been generated. The web services $WS_1$to WS_7 are the subset of available number of web services which satisfied the value of DeterminedQoS. From the set of available web services various paths are been analyzed which satisfies the service request, $Service_{Req}$ and the expected result, $Expected_{Res}$. In the below example, we could identify various paths, from which the best possible path would be selected in the next step of the theorem.

From the set of available web services, treating each web service as a node, various paths are identified based on the flow of information. Each path in the graph defines one composition for the service request. The graph and the path determination is done using existing AI algorithms. The paths of the graph are then represented in the form of linked list. Each entry in the list represents the composition of the service request.

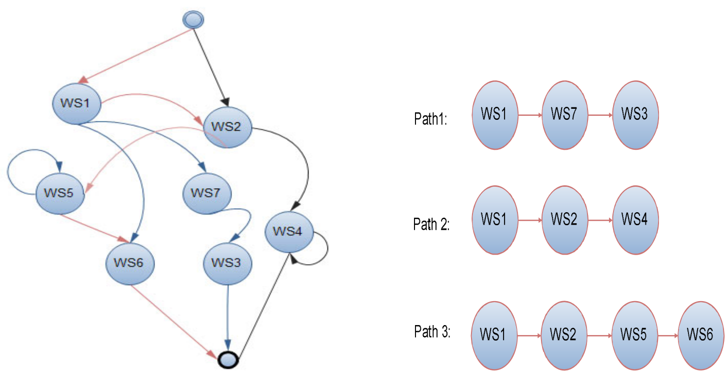

Fig. 2. A Sample graph of web service with the possible paths

Best Path Identification – Selecting the best composition for the service request

From the linked list representation arrived from task 2, the final task would be the identifying the best path and presenting the same to the requester. In identifying the

$$WS_c = \frac{\sum_{i=0}^{m} (AvailWS_i + CostWS_i + SucessRate\,WS_i + SecurityWS_i + TrustWS_i + Performanc\,eWS_i)}{m}$$

best, again the non functional parameter, the QoS, comes to picture. The QoS parameters are selected in par with the provider and the beneficent. The selection is based on the below given formula. if ‘n’ be the available set of web services which satisfy the 'DeterminedQoS', each web service composition would be comprised of a subset of web services say *m*.

3.3 Methodology

The selection of best composition from a set of paths identified from the graph is lies the effectiveness of composition. Initially the composition lies solely on the weightage given by the provider to the web service during registering in UDDI via WSDL file. Once the web service is consumed by the requester, based on their usage or through their explicit feedback on that candidate web service, the ranking among the other peers gets differed [13]. Upon using the web services, the requester registers a belief factor with respect to each candidate web service of the composition in the process registry. The belief factor is a numerical value. Higher belief value depicts the level of trust and reputation towards the web service. A feedback mechanism could be a questionnaire, likes etc [12][13].

In our proposed approach, an adaptive composition of web service based on the expected QoS is been analyzed. In the composition, the user might not be satisfied in a single web service. Say in our previous example from Fig 2, if WS_5 of path 3 alone is not reputed with respect to the user, a different composition just by changing that service alone would be devised, provided there is a similar kind of request in the near future. With minimal change in the WSDL file, a more qualified composition could be given in a minimal time.

The algorithm (algorithm 1) is designed to rank the various possible combinations of web service composition. CandidateWSList is the list of all web services those have been shortlisted, which satisfies the expected QoS requirements of the requester. DeterminedQoS is the expected QoS, which is been identified by from the preprocessing stage of composition. Weighted average of all the QoS properties of the candidate web service is been compared with its peers to identify the best composition. The first 'if' statement satisfies if there is a composition with a highest significance. In such a scenario, the composition with the highest QoS is declared as the best and been presented to the user.

Algorithm 1. QoS-Aware WSC (CandidateWSList, DeterminedQoS)

```
Initialize
    CompRank =0,
    CompToBePresented = 0.
    For each 'i' in CandidateWSList
    CompRank_i = Compute the weighted sum of QoS w.r.t the CandidateWSList_i
        If (CompRank_i > CompRank)
            CompRank = CompRank_i;
            CompositionToBePresented = i;
        End if
        If(CompRank_i = CompRank)
            Beleif_i = checkBeleif(i)
            Beleif_current = checkBeleif(CompToBePresented);
                If(Beleif_i > Beleif_current)
                    CompositionToBePresented = i;
                EndIf
        EndIf
        If(Beleif_i = Beleif_current) //Select arbitrarily
        EndIf
    End for
```

The second if statement would satisfy when two or more services are equal in terms both functional and non functional properties. The adaptive capacity of our system plays a major role. The belief of the corresponding web services with respect to the users is been analyzed and the option with the better belief is been offered. Lastly, if there is a scenario where we could not find a difference in any means from the functionality to the belief factor, we state that they both are equal and the best composition is been selected arbitrarily.

4 Experimental Results and Discussions

Our proposed QoS aware composition is analyzed for tourism domain. A sample ontology of tourism domain is been created. As discussed, the ontology also contains the QoS information of the candidate web services. The proposed approach is also verified against composition which has not considered QoS as a measure.

Considering a query related to tourism domain, the preprocessing phases of composition have identified, train, hotel, travels and restaurant as the candidate web service. The preprocessing step also includes the computation of 'DeterminedQoS' based on the request and the expected result. Those services that satisfy DeterminedQoS are selected. The various paths are been identified. Our scenario resulted in 5 compositions. Evaluation of the best composition both with and without QoS is been discussed below.

Without QoS Measure in determining composition

Each web service with respect to the functionality is been rated based on its functionality. This rating does not differ among the users. They stay constant and it's up to the provider or the registry to update the value.

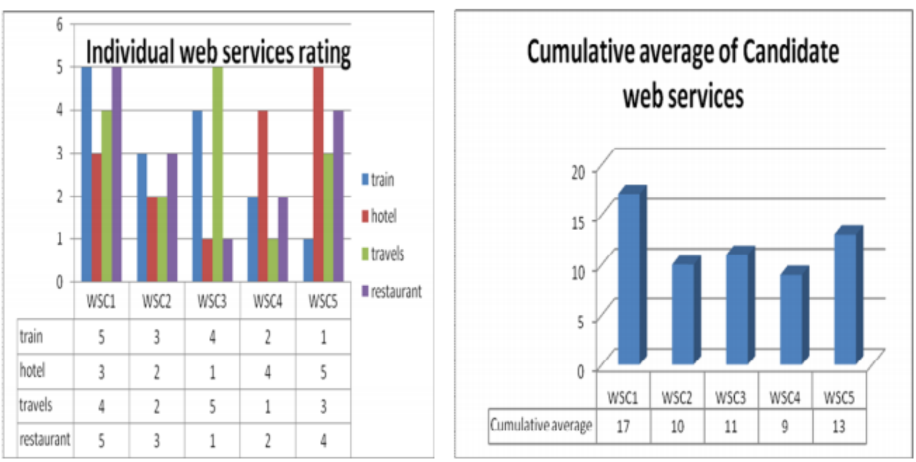

Fig. 3. Experimental result of composition without QoS paramaters

With QoS Measure in determining composition

In this approach, each web service is attached with QoS properties. The QoS properties are selected in concern with both the service provider and service requester. The QoS parameters considered with respect to provider are availability, cost, time and security. Trust and reputation are the factors been considered for requester. As there is a change in QoS properties between the design and run time, a change in composition is noticed.

From the Fig 4, it's noted that composition 3 is much better than composition 1 which has been analyzed at compile time. In the run time, based on the expected QoS based result, the WSDL file is been changed accordingly and composition is been provided in a short time span.

Comparative Study

There are various approaches defined for composition in literature. Table 1 summarizes the comparative study of those approaches with the proposed approach. The comparison is based on the features like service connectivity, the strategy used for composition, inclusion of QoS, the evaluation mechanism and graph representation. Our approach is to compose the web service based on both the semantic and QoS aspects of the web services and has all the features and so the approach is considered efficient when compared.

Table 1. Comparison of proposed framework with existing frameworks

Composition	Connectivity	Composition Strategy	QoS Modeling	Graph Support
Syntactic	Yes	Workflow	No	No
Semantic	Yes	Workflow + AI	No	Yes
QoS driven	Yes	Workflow	Yes	No
Semantic+ QoS driven	Yes	Workflow + AI	Yes	Yes

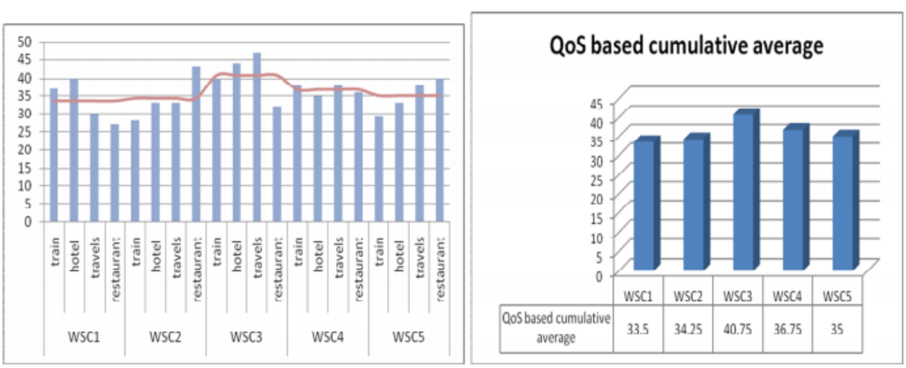

Fig. 4. Experimental result of composition with QoS paramaters

5 Concluding Remarks

With the existence of various attempts to web service composition, QoS aware web service paves way for the dynamic composition. The selection of services from the registry is would happen only if it satisfies the minimum expectation value stated by the customer. The degree to which the composition is efficient depends on the 'DetermineQoS' computation and the number of services in the registry which satisfies the computed 'DetermineQoS'. The use of a measure in the selection of web services makes a prominent selection and also minimizes the selection space. A separate module form monitoring the user's expectations namely the QoSMonitoring helps in identifying the expectation of the user at run time. Thus the paper discusses the importance and dynamic revision of QoS in the selection and composition of web services. The attempt to update the belief factor based on the feedback in the process registry and subsequently in the extended QoS-Domain ontology plays the major role in the adaptation of the web service composition at run time. As a future work, the addition of aspects in composing the web services is been proposed to get even finer result.

References

1. Huang, A.F.M., Lan, C.-W., Yang, S.J.H.: An optimal QoS-based Web service selection scheme. Information Sciences, 3309–3322 (2009)
2. Liu, M., Shen, W., Hao, Q., Yan, J.: An weighted ontology-based semantic similarity algorithm for web service. Expert Systems with Applications (2009)
3. Pastrana, J.L., Pimentel, E., Katrib, M.: QoS-enabled and self-adaptive connectors for WebServices composition and coordination. Computer Languages, Systems & Structures 37(1) (April 2011)
4. Wang, P., Chao, K.-M., Lo, C.-C., Huang, C.-L., Li, Y.: A Fuzzy Model for Selection of QoS-Aware Web Services. In: Proceedings of the IEEE International Conference on e-Business Engineering (ICEBE 2006), pp. 585–593 (2006)
5. Wang, P.: An Entropy Decision Model for Selection of QoS-Aware Services Provisioning. Department of MIS, Kun Shan University, Taiwan (2006)
6. Tran, V.X., Tsuji, H., Masuda, R.: A new QoS ontology and its QoSbased ranking algorithm for Web services. Journal on Simulation Modeling Practice and Theory, Science Direct 17(8), 1378–1398 (2009)
7. Toma, I., Foxvog, D., Jaeger, M.C.: Modeling QoS characteristics in WSMO. In: Proceedings of the First Workshop on Middleware for Service Oriented Computing, pp. 42–47 (2006)
8. Yu, T., Lin, K.-J.: Service Selection Algorithms for Web Services with End-to-End QoS Constraints. In: Proceedings of the 2005 IEEE International Conference on e-Technology, e-Commerce and e-Service, Hong Kong, China, pp. 129–136 (March 2005)
9. Liu, F., Lei, Z.-M.: Research on user-aware QoS based Web services composition. The Journal of China Universities of Posts and Telecommunications, Science Direct 16(5), 125–130 (2009)

10. Lv, C., Dou, W., Chen, J.: QoS-Aware Service Selection Using QDG for B2B Collaboration. In: Proceedings of the Fourteenth IEEE International Conference on Parallel and Distributed Systems, pp. 336–343 (2008)
11. Rong, W., Liu, K., Liang, L.: Personalized Web Service Ranking via User Group combining Association Rule. In: Proceedings of the IEEE International Conference on Web services, pp. 445–452 (2009)
12. Choi, S.W., Her, J.S., Kim, S.D.: QoS metrics for evaluating services from the perspective of service providers. In: Proc. IEEE Intl. Conf. on e-Business Engineering, pp. 622–625 (2007)
13. Papaioannou, I.V., et al.: A QoS ontology language for web services. In: Proceedings of 20th International Conference on Advanced Information Networking and Applications, pp. 18–25. IEEE Computer Society (2006)
14. Protégé, Ontology Editor (2007), http://protege.stanford.edu/
15. Luo, Y.-S., Qi, Y., Hou, D., Shen, L.-F., Chen, Y., Zhong, X.: A novel heuristic algorithm for QoS-aware end-to-end service composition. Science Direct (2010)