ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

Service Correlation and Constraint Aware Composite Service Composition under QOS Context

A.Goutami
M.Tech Student (CSE)
Kottam College of Engineering,
Chinnatekuru (V), Kallur (M),
Kurnool District-518218.

Mr.A.V.Rama Krishna Reddy
Assistant Professor
Kottam College of Engineering,
Chinnatekuru (V), Kallur (M),
Kurnool District-518218.

ABSTRACT:

The ability to perform web service discovery and composition automatically and dynamically is essential and has emerged as an important research topic. Automated web service composition deals with the significant increase in the number of available services over time, as well as frequent changes in their definitions. It enables significantly faster responses to user queries for composite services, compared to the manual case. Also, it produces compositions up-to-date with the latest web service definitions, despite the dynamic environment. A significant contribution of the approach concerns the full incorporation of semantics. The utilization of semantic information facilitates discovery and composition. It also permits approximate composition and enables the quality assessment of the produced composite services in terms of accuracy by using AI techniques especially planning. The framework maintains compatibility with the current standards, to ensure interoperability, and it is independent from specific planners. The model devised in this paper focuses on the addition of the OWL-S descriptions of produced composite services in the registry of available services, to explore the possibility to accelerate the composition process. In addition it deals with enhancing the approach with the ability to produce various composite services according to nonfunctional user preferences, dealing with pragmatic knowledge and As web service standards evolve, exploitation of pragmatic knowledge could be possible by extending existing web service description.

Introduction

Numerous procedures like accounting, eScience, finances, multimedia programs and supply chain management are shifting regarding the ad-hoc method for service composition. The composition concern gets most difficult with large amount of services obtainable that are growing each day and provide the same performance with variations of QoS. The selection of component services from a set of offered services usually results in complicated decision concern.

The purpose of a service selection algorithm is mapping of every single process chosen to performance to an applicable service from a set of obtainable services. This permits an optimized QOS in regards to the whole process and the user's specifications. In web services of compelling nature, the QoS assessments have concerns of deviations. This necessitates during execution decisions for selecting suitable algorithms as well as dynamic replacement of services in real-time (e.g. in multimedia programs) with no compromising the effectiveness.

A composition consult (e.g. in a workflow language like BPEL [1]) may be patterned as Multi Choice Multidimensional Knapsack (MMKP) issue that even so is an extremely difficult concern [2]. For an optimized QOS in regards to the overall process and the user's specifications to MMKP incurs massive cost.

The MILP (Mixed Integer Linear programming techniques) [3], have concerns of inadequate scalability as well as recent strategies [4, 5] cannot effectively maintain run-time specifications. In this

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

document an efficient and scalable heuristic technique for QoS-based service selection is evaluated with the following steps,

1. The complete QoS optimization concern is portioned into many sub-problems by native QoS optimization for complete effective solution.
2. The concern decomposition outcomes as explained in this document are applicable to a distributed design containing a service assembler also a group of distributed service agents.

The service selection according to QoS with the evaluated heuristic strategy cannot advise a certain optimal collection of services. Anyhow the need of the industry in provisions of resolution times, throughput and so on. are estimated solutions opposing exact remedies. Hence we need a practical number of services that provides most of the specifications with feasible costs and overcomes noticeable constraints assault. In this document the test assessments reveal the strategy operates compatible with all earlier techniques in solving concerns with challenging computations also concurrently provides quality outcomes.

Related Work

An extensible QoS computation design by Liu, Y., Ngu, A.H.H., Zeng, L [6] provides open as well as fair handling of QoS data, though cannot efficiently deal with QoS-based composition.

A strategy according to the selection of compelling as well as quality-driven services by Zeng, L., Benatallah, B., Dumas, M., Kalagnanam, J., Sheng, Q.Z [4] is according to utilizing global organizing for choosing the foremost service elements for the composition also utilizes strategies of (mixed) linear developing [3] for choosing the most outstanding component services.

Ardagna, D., Pernici, B [5] equivalent to the above strategy use linear programming model that is most effective in case of small size issues to incorporate local constraints. The search algorithms [8] utilized however has enormous time complexity impacting the scalability of the strategy.

A heuristic algorithm (WS_HEU) [7] for an effective strategy to choose a near-to-optimal remedy more than specific solutions has polynomial time complications. The algorithm reveals significant development against accurate solutions, anyhow fails regarding scalability as well as specifications of real-time rather than other web services increasing in number every day.

System Model and Problem Statement

Abstract vs. Concrete Composite Services

This model is according to the prediction of accessibility of a group of web services S described as a union of abstract service classes.

An abstract service class S_j where $S_j \in S$ denotes a group of web services of certain functionality. E.g. flight reservation web services of Lufthansa, Qantas and so on.

The strategy is according to the prediction that the data about service classes is handled by a group of service agents [6, 9]. Every web service according to a subscription program is capable of join or leave service classes at every point of time. The two aspects are classified as below,

- Any composite service, is illustrated as a composition demand as $CS_{abstract} = \{S_1, ..., S_n\}$. Here $CS_{abstract}$ signifies required service classes (e.g. flight reservation) and not any specified web service (e.g. Qantas flight reservation web Service).
- An efficient composite service, is an instant of an abstract composite service received by mapping every abstract service class in $CS_{abstract}$ to a definite web service s_j, such that $s_j \in S_j$.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

QoS Vector

In this document the quantitative non-functional attributes of web providers are regarded as measure the QoS. The characteristics are represented in the kind of a vector $Q_s = \{q_1, q_2, \ldots, q_r\}$.

The attributes regarded are i) Generic QoS features like availability, price, response time, reputation [6] and so on, and ii) Domain-specific QoS features like video quality for multimedia web services, bandwidth.

The features values specified to QoS are obtained from i) service suppliers, e.g. price, response time registered from monitoring the last execution, and so on. or from ii) user feedbacks, e.g. reputation.

The group of QoS features might be classified into two subsets i) positive features like throughput, availability and so on, whose values should be enhanced to their optimum value and ii) negative QoS features like price, response time and so on. whose values should be reduced to their minimal value. In this analysis only negative features are chosen for testing and for minimizing the complexity. Furthermore the positive features can be multiplied by -1 also converted into negative features. The function determines the i-th QoS parameter whereas in a class S of the web services, the service agent of the class deals with the details of QoS.

QoS Computation of Composite Services

The QoS value of a complex service is determined according to the QoS standards of its component services and the formulation method made use of (e.g. sequential, parallel, conditional and/or loops). In this analysis, we viewed as the provider selection algorithm for QoS-based service composition and its efficiency on the sequential composition model. The models that differ are changed to the sequential model by handling multiple execution paths and unfolding loops along with appropriate techniques [4].

The QoS vector for a composite service CS is revealed as $Q_{CS} = \{q_1'(CS), \ldots, q_r'(CS)\}$ where $q_1'(CS)$ reveals the estimated QoS values of a composite service CS built up from the estimated QoS values of its component services.

Utility Function

A utility function is utilized for finding the multi-dimensional excellence of a provided web service composition. For the utility function in this study a Multiple Attribute Decision Making strategy or the Simple Additive Weighting (SAW) strategy [10] is used.

The calculation of the utility function is completed by transforming of QoS characteristics values for a consistent measurement of the multi-dimensional service properties without concerning their units as well as ranges.

Afterwards a weighting procedure is applied to reveal the user dependent priorities as well as preferences.

The calculation in the scaling strategy requires every QoS characteristic value being transformed into a value around 0 and 1, by evaluating it with minimum as well as maximum feasible collected value that might be simply determined by aggregating the localized minimum (or maximum) feasible value of every service class in CS. For example, the maximum rendering price of any definite composite service is determined by accumulated the execution price of the more expensive service in every service class.

A Scalable QoS Computation

In this document we evaluate a scalable remedy for the issue of QoS-bases web service composition.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

Initially the global optimization concern is split into sub-problems i.e. mapping the global QoS calculation on the composite service level $U'(CS)$ into local computations for solving the problem on each service class individually. Second, an easy algorithm is used to divide each global QoS constraint $c_k^{'} \in C^{'}$ into n local constraints established locally on the component providers. Third and finally a allocated service selection algorithm uses local search for revealing global QoS specifications.

Fig 1: Shows the Scalable web service Composition

Decomposition of Global QoS Computation

The usage of (3) on the composite service standard needs specifying each conceivable combinations of the service prospects for ensuring the optimal selection. Anyhow the strategy is highly inefficient in case of big-scale issues or real time specifications of programs. To conquer this issue the $U'(CS)$ is customized into the utility function $U_{local}^{'}(s)$ also used on the component service level preventing evaluation of every possible combinations.

$$U'(CS) = \sum_{j=1}^{n}\sum_{k=1}^{r} \frac{Q\max(i,j) - qk(s_j)}{Q\max'(k) - Q\min'(k)}.w_k$$

Decomposition of Global Constraints

For a promise of local QoS computation outcomes satisfying the global QoS constraints, every global constraint $c_k^{'} \in C^{'}, 1 \le k \le m$ is divided into n localized constraints. The local statistics quality values are applied to determine an affordable decomposition of every global constraint $c_k^{'}$ as below,

First the localized constraint value c_{jk} of every service class is set to the localized maximum value of that class,

$$\forall c_k^{'} \in C^{'} : d_k = \sum_{j=1}^{n} c_{jk} - c_k^{'}$$

Distributed Optimization of the QoS Computation

An architecture containing a *service composer* and a quantity of *service brokers* - allotted or single machine is supposed. Every service broker handles the QoS data of a set of web service classes also preserves a set of existing web services combined with authorized specifications of their non-functional attributes or QoS features like response time, throughput, price and so on. The service composer starts a composite service CS by connecting with the service brokers.

$$\delta_k = \sum_{j=1}^{n} (c_{jk} - q_{jk}), 1 \le k \le m$$

Experimental Evaluation

In this research for assessing the model evaluated, several tests are performed as observe,

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

The model is considered by evaluating on a HP ProLiant DL380 G3 machine operating on Linux (CentOS release 5) as well as Java 1.6, with 2 Intel Xeon 2.80GHz processors also 6 GB RAM.

In this evaluation the efficiency and the quality of the outcomes of the model are reviewed with that of the linear programming techniques (LP) [4,5] also the heuristic algorithm WS_HEU [7]. In scenario of linear programming techniques, open source Linear Programming method Ipsolve version 5.5 [11] is utilized. In scenario of WS_HEU an own execution is used. The execution is completed practically regarding each possible optimizations minimizing to the low the time of calculation. The assessments are performed with several situations of the QoS composition issue with modifications of the amount of service classes (n) also the amount of service applicants per class l. These constraints have unique pairing and the specific combination is displayed as one illustration of the composition concern.

The QoS information the QWS real dataset in [12] is applied. The dataset consists of 9 QoS features specifications relevant to 364 real web providers. For an evaluation of the scalability issue a test according to bigger group of services is needed. To conquer this issue, the QWS dataset is tested on many times, multiplying the values of web providers' quality every time with a consistently dispensed random value between 0.1 as well as 2.0. With this strategy it is prospective of getting a data set of concerning 100.000 services.

Performance Evaluation

The efficiency assessment of the preceding three techniques is completed by evaluating the time appropriate for choosing the remedy or the best pairing of tangible services of specific methods.

In Fig 2, Performance analysis of automated service trace and composition strategies under composition accuracy metric, WS_HEU also the model we recommended DISTHEU is revealed. In our studies the efficiency of the three strategies is considered in regards to the size of the issue, amount of service classes (n) as well as the amount of service prospects per class *l*.

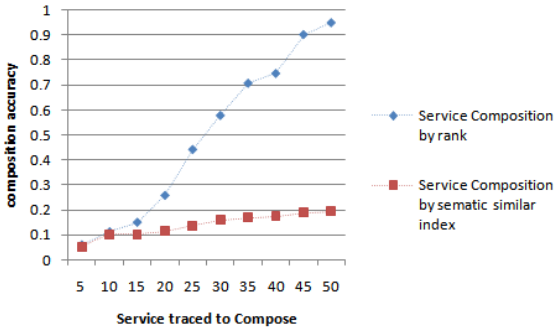

Figure 2: Performance analysis of automated service trace and composition strategies under composition accuracy metric

Optimality Evaluation

The Optimality is estimated with the optimality ration $R = {}^{Ua}up$. A review of the optimal outcomes of the strategy evaluated with that of LP technique is completed. The strategy evaluated in this document produces the utility of the best composition U_{approx} as per (3) also the LP technique produces the utility of composition U_{opt}. The outcomes revealed in fig 3 illustrate that DIST_HEU reaches best outcomes with an average of 98% optimality ratio. Furthermore the quality of the outcomes of the evaluated strategy DIST_HEU decreases by average, simply 1% below the outcomes of WS_HEU. The cost included is extremely high however for WS_HEU for this small calculation time advancement as observed.

Conclusion and Future Work

The efficient web service discovery needs, the user must be able to discover all appropriate web services within the UDDI irrespective of the predefined categories, and all appropriate web services must be successfully discovered even if the user is not aware of all the relevant terms that include all appropriate web services. In this paper we have considered the semantic

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

based ontology approach involves service categorization and selection of services with semantic service description and the composition of web service using OWL-S. In this regard, this work tends to actuate the requirement to integrate automated service composition. . We have tested the proposed approach by using a sample web service application. As future work, we extend to explore additional mapping tools to express service request to search for relevant concepts.

References

1. OASIS: Web services business process execution language (April 2007) http://docs.oasis-open.org/wsbpel/2.0/wsbpel-v2.0.pdf.

2. Pisinger, D.: Algorithms for Knapsack Problems. PhD thesis, University of Copenhagen,Dept. of Computer Science (1995)

3. Nemhauser, G.L., Wolsey, L.A.: Integer and Combinatorial Optimization. Wiley-Interscience, New York, NY, USA (1988)

4. Zeng, L., Benatallah, B., Dumas, M., Kalagnanam, J., Sheng, Q.Z.: Quality driven webservices composition. In: WWW. (2003) 411–421

5. Ardagna, D., Pernici, B.: Adaptive service composition in flexible processes. IEEE Trans.Software Eng. 33(6) (2007) 369–384

6. Liu, Y., Ngu, A.H.H., Zeng, L.: Qos computation and policing in dynamic web serviceselection. In: WWW. (2004) 66–73

7. Yu, T., Zhang, Y., Lin, K.J.: Efficient algorithms for web services selection with end-to-endqos constraints. ACM Trans. Web 1(1) (2007)

8. Maros, I.: Computational Techniques of the Simplex Method. Springer (2003)

9. Li, F., Yang, F., Shuang, K., Su, S.: Q-peer: A decentralized qos registry architecture for webservices. In: ICSOC. (2007) 145–156

10. Yoon, K..P., Hwang, C.L.: Multiple Attribute Decision Making: An Introduction (QuantitativeApplications in the Social Sciences. Sage Publications (1995)

11. Michel Berkelaar, Kjell Eikland, P.N.: Open source (mixed-integer) linear programmingsystem. Sourceforge http://lpsolve.sourceforge.net/

12. Al-Masri, E.,Mahmoud, Q.H.: Investigating web services on the world wide web. In:WWW.(2008)