ISSN: 2229-6956 (ONLINE)
DOI: 10.21917/ijsc.2017.0228

AN INNOVATIVE FRAMEWORK FOR AUTO DYNAMIC SELECTION AND COMPOSITION OF WEB SERVICES

Khozema Shabbar[1], Tarun Shrimali[2] and Manjur Kolhar[3]
[1]*Department of Computer Science and Engineering, School of Engineering and Technology, Career Point University, India*
[2]*Sunrise Group of Institutions, India*
[3]*Department of Computer Science, Prince Sattam Bin Abdulaziz University, Saudi Arabia*

Abstract

With the availability of ample amount of Web Services similar in functionality and varying in Quality of Service (QoS) over the web, it is essentially required to get the best-suited candidate services for the resultant composite service. QoS plays a decisive role in selecting web Services individually and Composite Web Service as a whole. Certainly, composition is one of the key properties of service-orientation to create new as well as advanced level services, by re-using the existing ones. In distributed environments, services having no quality guarantees, adversely affects the composition outcome. To find the best candidate services for the composition, researchers have opted either Users' feedback or Providers' published information but none of them have considered both of them together. In our approach, we have taken into consideration both the things (Users' feedback and Providers' published information) together in order to find the best candidate web services by cross verifying both of the information to ascertain the quality of the candidate web services. An innovative framework for the composition of web services along with the detailed methodology and algorithm has been discussed in the second half of the paper.

Keywords:

Web Services Composition, Web Services Selection, Web Services, Users' feedback, Providers' published information

1. INTRODUCTION

In the age of digital computing, Web services are gaining greater acceptance as far as e-commerce and distributed computing are concerned. As web services are tied loosely thus paving the way for the producers to produce and compose them at run time too. The process of clubbing together atomic or candidate web services in order to construct value-added Web applications and services is known as Web Service Composition and the resultant web service is known as composite web service. QoS related information provided by Publishers' may not always be true and Users' feedback may also not be correct or say biased. Hence rather depending on any one only, we have considered both of them in our approach and proposed a framework to compose web services taking into account both Users' feedback and Providers' published information. Both of the information are cross verified to ascertain the quality of the candidate web services prior to performing composition. Detailed methodology and functionality are presented in this paper.

The paper is composed of various sections. Section 2 is the related work. Section 3 is the preliminaries. Section 4 is the proposed composition framework. Section 5 is the proposed technique. Section 6 is the proposed implementation and evaluation. Section 7 is the conclusion and future work.

2. RELATED WORK

Though, dynamic composition executed at run time is considered to be a complicated task but the key advantage is - discovery and invocation of web services on the users' demand dynamically. Examples related to dynamic compositions are - intelligence algorithm, Semantic Web, Negotiation etc. etc. Some of the works related to web services dynamic composition are summarized here under.

Garg et al. [1] introduced two things: (1) a novel method of composition for Semantic web services based on Multi-Agent System and (2) An innovative approach to select those semantic web service providers, to be included during the course of composition. Moreover, they have discussed important issues like proper selection model, input validations, negotiation conditions etc. Also, they have discussed some of the implementation issues besides presenting an evaluation and comparison of the existing works.

Rong et al. [2] presented QoS aware web service composition mechanism based on pragmatic agent presenting a new viewpoint over web service composition. They have discussed the limitations of the semantic web services and come up with the concept of the Pragmatic web i.e. the extended and complemented version (of semantic web) enabling negotiation, communication and intension among service customers and service providers in service oriented architecture for the higher quality of cooperation and service orientation.

Xiaotang et al. [3] offered a QoS aware composition model for web services (WSCMQC) capable of attaining optimal QoS aware Web service workflow combinations having no restraint.

Jing et al. [4] presented Composition model for Web Services explaining the way to achieve more precise results by monitoring agent-based composition at execution time.

Siala et al. [5] introduced architecture based on Multi Agents to determine the optimized Composite QOS (CQoS). Negotiation with Multiple Agents played vital role in selecting Composite Web services. While performing web service composition, users' preferences were given highest priority. Agents were then put for negotiating the QoS value and lastly, dynamic selection of Service Providers, providing different services, took place in the composition. Important factor over here is that negotiation was taking place with only available Web services providers which resulted in an improved CPU time.

Yuan-sheng et al. [6] introduced and implemented a QoS-aware composition framework for Web services. Proposed framework supported both the concept of Visualization modeling as well as the method of dynamic web service selection. Overall structure and operational mechanism was discussed in detail focusing on process design module and publish module of the web services. Also, they have executed global optimization algorithms to achieve selection of the suitable services dynamically that meet both existing services QoS as well as composite web service QoS.

Rang-sheng et al. [7] introduced QoS-based model that performs composition of Web Services dynamically, well-addressing the core issues of selection and coordination. They have presented an algorithm (WS_TSC) taking into account various constraints e.g. response time, rate of success and rate of composition, for the selection of services. Finally, they conducted simulation experiments to evaluate the dynamic Web service composition performance and obtained improved results with respect to response-time, success-rate and composition-rate.

Lu Li et al. [8] introduced QoS aware Multi-dimension architecture for the selection and composition of web services. The purpose of introducing the idea of Multi-dimension is to describe web service composition QoS constraints. They calculated the overall QoS value of web service composition based upon the type of web service composition. At the end, the Web service composition, satisfying user’s non-functional constraints optimally (along with satisfying functional constraints) was selected as the final outcome amongst others.

Farhan et al. [9] presented a unique approach to carry out web services auto-dynamic composition clubbing together both interface based and functionality based approaches. They tried to handle problems pertaining to data distribution, QoS and execution etc. They have concentrated mainly to solve the problems related to decentralized dataflow and presented a mechanism that resulted in better response time, minimal latency, and highest throughput. Authors have also suggested solution to get rid of the problems like - frequent changes in parametric values (input as well as output values), network related issues and independent nature of web services, which occur during the composition of web services.

Ming-Wei et al. [10] introduced a pragmatic approach to perform workflow-based web service composition addressing the issue of the relations between the web service QoS and environments which in general is neglected. To address the issue, they have presented a composition approach based on Production-QoS-Rule & applied ‘black box’ analysis technique to retrieve optimized composite service. They first recorded Execution information of composite services in order to be used as a basis for the subsequent QoS knowledge mining and statistical analysis. They computed QoS values of Web services in a constant interval of time and mined the Production-QoS-Rules (applied to represent varied QoS performance of Web services qualitatively that occur in dissimilar environments). Finally, optimized composite service is selected based on resultant QoS knowledge of Web services.

Liu et al. [11] proposed a credible mechanism of web service composition incorporating two things - QoS dynamic prediction and global QoS attributes decomposition. The main feature of the method is that initially for each service class they are decomposing global QoS attributes into local ones and then transforming the issue of dynamic composition as a localized optimization issue. Also, authors are applying improved case based reasoning to predict and ascertain QoS values of the candidate web services prior to selecting them for the final composition.

Karunamurthy et al. [12] presented a different method for web services composition that comprises three building blocks namely - composition framework, description framework and business model. The composition framework supports all of the characteristics including - non-functional, functional, semantic and behavioral characteristics of web services. The description framework incorporates all of the four characteristics of Web services that enable the description of Web services and provide the languages needed for describing the Web services in both the composition framework and the business model. The business model is composed of three entities: web service provider, web service composition registry and web service composer. The Web service composer is accountable of the overall composition process that is well-noticed by the composition framework.

Mehdi et al. [13] proposed a multi agent architecture for composing the web services dynamically based on communities and introduced a solution for the failed candidate service agent substituting it by another equivalent one from the same community in order to ensure higher availability of candidate services in composition process. Proposed architecture was implemented to validate its two features - dynamic composition and the replacement of a failured service by another one.

Hammas et al. [14] introduced an architecture for the composition that has two features: A) Dynamic Selection that means tying of candidate services at the time of execution and B) Adaptive Composition that means having an updated information about the status of candidate service at the time of execution and replacing of a service by another one in case of failure. Proposed architecture supports global QoS optimization introducing Ant Colony based optimization algorithm.

Freddy et al. [15] presented architecture for dynamic web service composition applying the concept of semantic matchmaking between the two service parameters: inputs and outputs in order to facilitate their interaction and interconnection. Semantic matchmaking facilitates in finding semantic compatibilities between service-descriptions defined independently. Also, they have presented a composition algorithm based on semantic graph in which service compositions are represented by a graph and semantic relations among services, by the nodes of the graph. Furthermore, both the attributes (functional & non-functional) of services are used to facilitate the computation of the best suitable and significant service composition.

Wang et al. [16] proposed an innovative recommendation method to compose the web services computing QoS-credibility of each of the web service units. Thereafter, weight of users’ preference is combined with the value of credible-QoS to compute the web service evaluation result.

Shahrokh et al. [17] have introduced an algorithm by combining heuristic and genetic algorithms together resulting in a semi-heuristic genetic algorithm. This semi-heuristic approach alters chromosomes depending upon the conditions, not satisfied. Research outcomes proved that the proposed method can be implemented satisfying users’ requirements more proficiently than other ones. The efficiency of the algorithm lies in low computation time which resulted in improving computation time.

Kashyap et al. [18] presented QoS aware composition mechanism introducing a Membership-function. This Membership-function prioritizes component web services which are similar in functionality to a greater level depending on their response time paving the way to be incorporated in the process of composition, increasing users' satisfaction.

Zhang et al. [19] introduced a QoS aware innovative approach for dynamic -composition and optimization of web services using Ant-Colony-Optimization algorithm (ACO). For dynamic composition, a multi-objective optimal-path selection model was designed and an updated form of *Ant-Colony-Optimization algorithm* was offered to solve the issue related to *multi-objective-optimization*.

Silva et al. [20] proposed a (graph-based) Particle-Swarm-Optimization method for the selection and composition of web services ensuring an optimal workflow & near-optimal web service selection in QoS aware service composition. Authors tried to address various constraints of the existing PSO-based approaches as the proposed approach does not require any workflow configuration and doesn't depend upon the users who have domain expertise. Also, the final outcomes of proposed method remarkably confirmed its better performance over greedy based PSO technique.

A plethora of research work discussed and suggested various frameworks/architectures for the selection and composition of web services based on Work flow, Multi-agent negotiation, Multi-dimension QoS, Ant Colony optimization etc. etc. but none of them focused on Users' feedback and its cross verification with providers' published information and vice versa. In our proposed framework, we are incorporating this important factor introducing an engine called – Information Verifying Engine (IVE). The same is discussed in detail in the below section.

In this study, we propose IVE for web services. To the best of our knowledge, no other study has inspected the mechanism of service allocation on the basis of review report of the services. Our initial study divulges that our proposed framework is capable of allocation of web services dynamically based on different customer requirements.

Drawback of our frame work is related to downtime. Our frame work is based on Internet, it means access to our framework will be inclined by entirely on the backhaul system. Another drawback of our framework is that it is not proposing anything on security. This, of course, poses a problem especially with respect to web services.

3. PRELIMINARIES

3.1 WEB SERVICES COMPOSITION

Web services are termed as distributed applications available over the Internet and are accessible through standard protocols such as SOAP. Not like the applications where static binding is a prerequisite before execution and discovery, the key feature of the web services lies in their discovery and invocation dynamically on users' request. Semantic and ontological techniques, in particular proved to be important in discovering and composition of web services dynamically altogether at the execution time.

3.2 AUTO-COMPOSITION OF WEB SERVICES

Automated composition approaches are mostly AI based approaches that can automatically generate the request and response. First of all, Translator translates a users' request from a standard form to a system adaptable form, and then those services which are fulfilling users' criteria are chosen from the repositories. Process Generator starts process of composing these chosen services and if more than one composite services meeting users' criteria then it is the Evaluator which evaluates all of them based on QoS and sends the best selected services to Execution Engine for final composition. Final outcome is returned to the Requester. A comprehensive dynamic composition Mechanism is presented in Fig.1 [21].

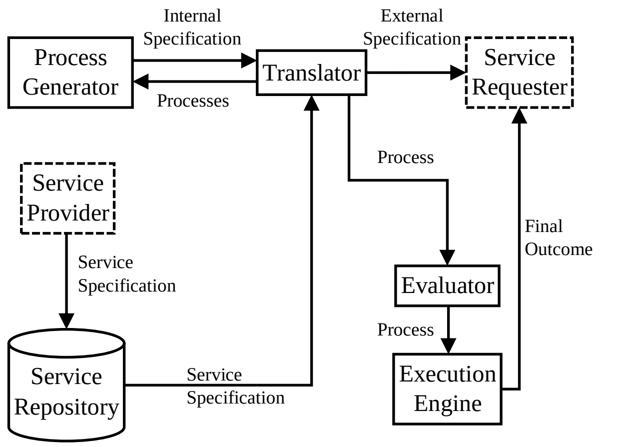

Fig.1. Generalized Dynamic Composition Mechanism

3.3 THE REPUTATION MEASUREMENT

The word reputation stands for an overall perception users' have about particular Web services. In other words, the reputation is users' overall feedback ratings about the services used or interacted by the users' in the past. Feedback rating is the evaluation by the users about the services invoked by them. It could be expressed in the form of a single value or a vector where a single value represents an overall perception and a vector represents a value for every QoS property of a web service like availability, response time and reliability. There are two important aspects to be considered while taking into consideration the users feedback,

- It is hard to ascertain the pureness of feedback ratings provided by the users since there exists malicious users. Malicious users may for commercial benefits, give malicious feedback ratings that wrongly impact on the measuring-results. Therefore, malicious users start flooding malicious feedback ratings which can subvert the Web services reputation system.
- We find various types of users over the net. They in general give dissimilar feedback ratings for the same service as per their personal expectations and opinions.

Similarly, there exists some service providers, not honest in nature and misuse the system. They sometime, furnish fake or wrong information in order to improve the probability of some of the candidate services to get selected or to reduce the probabilities of others in the composition process. They sometime, may fail even providing the published promised/ quality at execution time. Hence, the reputation of Web services need to be cross verified (considering both Users' feedback & providers' published information) while doing a service selection.

3.3.1 Malicious Rating Detection:

Authors [22] have applied the Cumulative Sum Method (CUSUM) to find out and check out feedback ratings malicious in nature. The CUSUM being a sequential analysis technique developed for identically distributed independent random variables is generally used for detecting changes based on hypothesis testing. Let say, for a process $\{y_i\}$ (i = 1,2,...), two hypotheses, θ_0 and θ_1, exist having respective probability density functions $p\theta_0(y_i)$ and $p\theta_1(y_i)$. These two hypotheses correspond to the statistical distribution takes place prior and later of a change respectively. The CUSUM for marking a change depends upon the log-likelihood ratio C_n and can be presented as,

$$C_n = \sum_{i=1}^{n} c_i \tag{1}$$

where, $c_i = \ln \dfrac{p\theta_1(y_i)}{p\theta_0(y_i)}$.

The general trend of the log-likelihood ratio comprises a negative drift and a positive drift before and after the change respectively. Thus, the significant information for identifying a change is found by getting the difference of value between the log-likelihood ratio, C_n (n = 1, 2,...), and a threshold value h(h > 0). If $C_n \geq h$ i.e. a positive shift takes places in the n^{th} sample leading to an abnormal detection point. CUSUM is best-suited to check the abnormal shifting and has widely been in use in detecting mini to moderate mean shifting.

3.3.2 Rating Adjustment:

Since feedback ratings done by the malicious users can be identified using CUSUM but feedback ratings are mostly dependent upon the different opinions expressed by the users for the same service leading to inaccuracy of the feedback ratings or say fails in ensuring the accuracy of the feedback ratings. As there exists varieties of users over the Internet with different preferences/opinions, feedback ratings reported by them solely depend upon their opinions.

Some of the users may be aggressive, others may be neutral or conservative. Henceforth, it is so vital to prevent the influence of aggressive, neutral or conservative feedback ratings done for the same service.

We are proposing here feedback similarity computation to prevent the influence of different opinions expressed by the users and to adjust their feedback ratings using the PCC (Pearson Correlation Coefficient). We take m number of users and n number of Web services and the relationship between them is denoted by a matrix of size $m \times n$. Each entry $r_{a,i}$ in the matrix represents the feedback rating done by the user a for the Web service i. $r_{a,i}$ here stands for a normal feedback rating.

Following is the equation used by the PCC in order to compute the similarity between users a and u depending upon their commonly-rated Web services:

$$Sim(a,u) = \frac{\sum_{i \in I_a \cap I_u} \left(r_{a,i} - \overline{r}_a\right)\left(r_{u,i} - \overline{r}_u\right)}{\sqrt{\sum_{i \in I_a \cap I_u} \left(r_{a,i} - \overline{r}_a\right)^2} \sqrt{\sum_{i \in I_a \cap I_u} \left(r_{u,i} - \overline{r}_u\right)^2}} \tag{2}$$

Here $Sim(a,u) \in [-1, 1]$ denotes the similarity between the users a and u where bigger value represents greater similarity, $I_a \cap I_u$ indicates a set of web services commonly rated by the users a and u, $r_{a,i}$ and $r_{u,i}$ are feedback ratings done by users a and u for the same web service i and $\overline{r}_a$, $\overline{r}_u$ represent the average feedback rating by the users a and u for all of the web services.

3.3.3 Discarding or Preventing Malicious Rating:

The core idea behind preventing malicious feedback ratings is to find the IP addresses associated with malicious feedback ratings and then filter them out using standard 'Bloom_ filter'. A standard Bloom_ filter is defined as an array of m bits which represents a set of n elements ($S = \{x_1, x_2,...,x_n\}$), initializing them all to zero. A standard Bloom_ filter takes k different hash functions - $h_1, h_2,...,h_k$ each of them maps some set element to one of the positions of m array with a uniform random distribution over the range 1,2,...,m. For each element $x \in S$, the bits $h_j(x)$ are set to 1 for $1 \leq j \leq k$. To check whether an item y belongs to S or not, we check the value of all $h_j(y)$. If they all are set to 1 means y is a member of S otherwise y is not a member of S. In case, if all $h_j(y)$ bits are set to be 1 and y is not a member of S, certainly it is a false positive. In summary, we can say that the key of preventing malicious feedback rating lies in identifying the IP addresses associated and disabling them from rating web services.

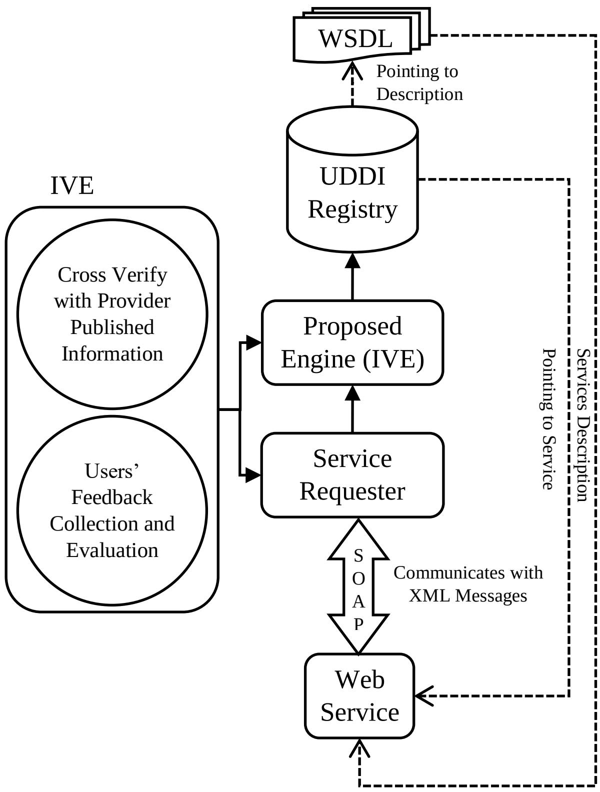

Fig.2. Enhanced Web Service Framework

3.4 ENHANCED WEB SERVICE FRAMEWORK

The Proposed Information Verification Engine (IVE) (shown in Fig.2) comprises two sections: feedback collection and evaluation section and another is verification section.

4. PROPOSED COMPOSITION FRAMEWORK

Various components of the proposed Composition framework are depicted in Fig.3 and are discussed here under -

1. **Web Services Registration**: The method of specifying web services to the system is termed as service registration. New services are enrolled in the service-registry through this process. Service providers maintain various service registries and are synchronized regularly in a course of time.
2. **Service Requester (SR)**: Requesters through service request module send request for a particular service.
3. **Translator (T)**: Translator's job is to translate user's request from an external form to a system adaptable form and to translate response conversely.
4. **Web Server**: Registries are hosted on server over the web.
5. **Information Verification Engine (IVE)**: IVE is an engine that cross verifies selected web services (meeting users' stated criteria) with the help of Users' feedback and Providers' published information before returning them for the final composition.
6. **Web**: Here in the framework, Web stands for WWW network where service providers do register their web services in UDDI registries.
7. **Composer (C)**: The job of the composer is to perform composition of the selected candidate services. The resultant web service is then called as composite web service.
8. **Service Engine (SE)**: The job of service engine is to check for the web services from the web meeting users' request and returns those to the composer.
9. **Service Registry**: Service providers' published web services are registered in the service registries and are used to make request for user's preferred web services. References of the web services being hosted on the service repositories are maintained by each registry.

5. PROPOSED METHOD

5.1 PROCEDURE

Steps of the procedure of the proposed framework are discussed here under:

- Very first registration of Web services take place in various registries.
- User makes request for a particular service.
- Translator translates users' request into a system adaptable form.
- The request reaches to the Composition Unit where Service Engine checks for users' desired service from multiple registries lying on the web returning results to the IVE.
- IVE verifies these selected candidate web services in two stages. In the first stage, it sends the selected web services to seek Users' feedback, whereas in second stage it cross verifies Users' feedback with Provides' published information and vice versa. After verification, verified services are sent to composer.
- Composer performs composition of these verified services and sends result to Execution Engine (EE) which in turn executes these web services. Final outcomes are returned through translator back to the user.

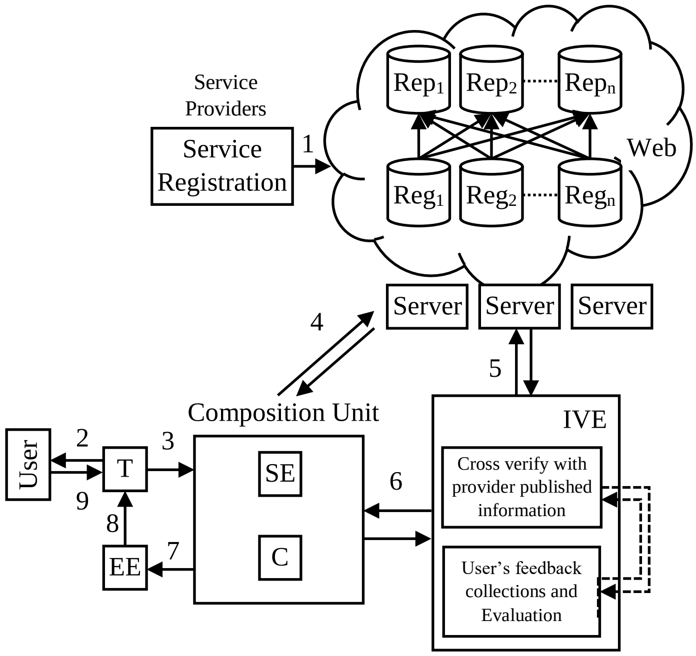

Fig.3. Proposed Web Services Composition Framework

5.2 PSEUDO CODING

Pseudo coding of the procedure involved is outlined here under:

Algorithmic steps of Web services composition –

Input: Users' Request for specified service

Output: Composite web service

Step 1: Web services are registered in the web registries;

Step 2: Translator does translate users' request;

Step 3: User makes request for desired service;

Step 4: Request arrives at Service engine;

4.1 Service engine searches Web UDDI registries for service;

4.2 Selects matching services;

4.3 IVE verifies the selected matching services;

4.4 Services based on Users' Feedback and Providers' Published Information;

4.5 Composer does composition of verified services;

4.6 Composer returns outcome to service Engine;

4.7 Service engine in turn returns outcome to Execution Engine;

4.8 Execution engine executes these services and final outcomes are returned to the User through

4.9 Translator

6. PROPOSED IMPLEMENTATION AND EVALUATION

An innovative Service Composition Framework has been introduced in this paper incorporating an Information Verification Engine (IVE). Proposed IVE contains two modules: 1) Users' Feedback module and 2) Cross Verification Module (with Providers' Published Information). Users' Feedback Module will be implemented with the help of Statistical Methods. In the first step, Users' feedbacks will be collected and malicious (fake) feedbacks will be detected using Cumulative Sum Method (CUSUM). In the second step, feedback ratings will be adjusted using Pearson Correlation Coefficient (PCC). In the third and last step, malicious ratings/ feedbacks will be filtered out using standard Bloom filter. Resultant filtered users' feedbacks will finally be cross verified with the Providers' published information in order to select the best candidate services for the composition. Since the communication between these two modules will take place in the form of XML messages, java programming language will be used to implement the proposed technique.

The aforementioned statistical equations for users' feedback evaluation will be implemented using SPSS version 24 and an actual feedback rating data set from a real online dating service (Libimseti) [23].

7. CONCLUSION AND FUTURE WORK

The major contribution of this paper lies in presenting an innovative framework for automated and dynamic web service composition by incorporating an Information Verification Engine (IVE) which not only does take into account the factual Users' feedback for selecting component web services but also does cross verification with the Providers' published information and vice versa. In our study, it has been found that authors in their works have more or less relied either on Users' feedback or Providers' provided information which led to wrong selection of component web services as Providers' published information may not always be trustworthy and updated similarly users' feedback also may not always be correct and fair. That's why in our approach Providers' published information and Users' feedbacks about web services are cross verified in order to ensure the selection of best component services for the composition. As a future work, proposed IV Engine will be implemented using SPSS and Java programming language.

REFERENCES

[1] Sandeep Kumar Garg and R.B. Mishra, "A QoS Aware, Multi-Agent based Approach to Semantic Web Service", *Proceedings of International Symposium on Information Technology*, pp. 12-16, 2008.

[2] Rong Weng, Lin Liang and Kecheng Liu, "Pragmatic Agent Based QoS Aware Web Services Composition", *Proceedings of 3rd IEEE International Conference on Semantics, Knowledge and Grid*, pp. 507-510, 2007.

[3] Xiaotang Li, Zhan Feng and Zhan Shaobin, "QoS-based Web Service Composition Technology Research", *Proceedings of International Conference on Computer and Automation Engineering*, pp. 601-604, 2010.

[4] Chen Jing and Wen-Bo Xu, "Web Service Composition Based on QoS Control Technique", *Proceedings of 9th IEEE International Symposium on Distributed Computing and Applications to Business, Engineering and Science*, pp. 281-284, 2010.

[5] Fatma Siala and Khaled Ghedira, "A Multi-Agent Selection of Web Service Providers Driven by Composite QoS", *Proceedings of International Symposium on Computers and Communications*, pp. 55-60, 2011.

[6] Yuan-Sheng Lou, Zhen-Hong Tao, Lu-Lu Tao, Hong-Tao Xu, Zhi-Hong Xi and Zhi-Feng Wu, "A QoS-based Web Service Dynamic Composition Framework", *Proceedings of 9th IEEE International Symposium on Distributed Computing and Applications to Business, Engineering and Science*, pp. 188-192, 2010.

[7] Rang-Sheng Dong, Wang Fei-Ming and Luo Xiang-Yu, "Dynamic Web Services Composition Based on QoS Model", *Proceedings of International Conference on Intelligent Computing and Integrated Systems*, pp. 823-826, 2010.

[8] Lu Li, Mei Rong, Guang Quan Zhang, "A Web Service Composition Selection Approach based on Multi-Dimension QoS", *Proceedings of 8th IEEE International Conference on Computer Science and Education*, pp. 1463-1468, 2013.

[9] Farhan Hassan Khan, M.Younus Javed, Saba Bashir, Aihab Khan and Malik Sikandar Hayat Khiyal, "QoS Based Dynamic Web Services Composition and Execution", *International Journal of Computer Science and Information Security*, Vol. 7, No. 2, pp. 147-152, 2010.

[10] Ming-Wei Zhang, Bin Zhang, Ying Liu, Jun Na and Zhi-Liang Zhu, "Web Service Composition based on QoS Rules", *Journal of Computer Science and Technology*, Vol. 25, No. 6, pp. 1143-1156, 2010.

[11] Zhi Zhong Liu, Zong Pu Jia, Xiao Xue, Ji Yu An, "Reliable Web Service Composition based on QoS Dynamic Prediction", *Soft Computing*, Vol. 19, No. 5, pp. 1409-1425, July, 2014.

[12] Rajesh Karunamurthy, Ferhat Khendek and Roch H. Glitho, "A Novel Architecture for Web Service Composition", *Journal of Network and Computer Applications*, Vol. 35, No. 2012, pp. 787-802, 2011.

[13] Sabrina Mehdi and Nacer Eddine Zarour, "Composition of Web Services using Multi Agent based Planning with High Availability of Web Services", *Proceedings of 2nd IEEE International Conference on Advanced Technologies for Signal and Image Processing*, pp. 10-15, 2016.

[14] Olfa Hammas, Saloua Ben Yahia and Samir Ben Ahmed, "Adaptive Web Service Composition Insuring Global QoS Optimization", *Proceedings of International Symposium on Networks, Computers and Communications*, pp. 23-26, 2015.

[15] Freddy Lecue, Eduardo Silva and Luis Ferreira Pires, "A Framework for Dynamic Web Services Composition", Available at: http://ceur-ws.org/Vol-313/paper4.pdf

[16] Wang Denghui, Huang Hao and Xie Changsheng, "A Novel Web Service Composition Recommendation Approach based on Reliable QoS", *Proceedings of IEEE 8th International Conference on Networking, Architecture and Storage*, pp. 321-325, 2013.

[17] Pooya Shahrokh and Faramarz Safi-Esfahani, "QoS-based Web Service Composition Applying an Improved Genetic Algorithm (IGA) Method", *International Journal of Enterprise Information Systems*, Vol. 12, No. 3, pp. 60-77, 2016.

[18] Namrata Kashyap and Kirti Tyagi, "Dynamic Composition of Web Services based on Qos Parameters using Fuzzy Logic", *Proceedings of International Conference on Advances in Computer Engineering and Applications*, pp. 778-782, 2015.

[19] Wei Zhang, Carl K. Chang, Taiming Feng and Hsin-Yi Jiang, "QoS-based Dynamic Web Service Composition with Ant Colony Optimization", *Proceedings of IEEE 34th Annual Computer Software and Applications Conference*, pp. 493-502, 2010.

[20] Alexandre Sawczuk da Silva, Hui Ma and Mengjie Zhang, "A Graph-based Particle Swarm Optimisation Approach to QoS-Aware Web Service Composition and Selection", *Proceedings of IEEE Congress on Evolutionary Computation*, pp. 3127-3134, 2014.

[21] Khozema Ali Shabbar, Tarun Shrimali and Mohammed Sha, "Computation of QoS While Composing Web Services", *International Journal of Advanced Computer Science and Applications*, Vol. 8, No. 3, pp. 309-317, 2017.

[22] Shangguang Wang, Zibin Zheng, Zhengping Wu, Fangchun Yang and Michael R. Lyu, "Reputation Measurement and Malicious Feedback Rating Prevention in Web Service Recommendation Systems", *IEEE Transactions on Services Computing*, Vol. 8, No. 5, pp. 1-14, 2014.

[23] L. Brozovsky and V. Petricek, "Recommender System for Online Dating Service", *Proceedings of 6th Conference on Znalosti*, pp. 1-12, 2007.