TABLE OF CONTENT

LIST OF FIGURES

LIST OF TABLES

CHAPTER – 1

1 INTRODUCTION

1.1 PROLOGUE

The applications for a business or industry for the integration of its business process within and across verticals and various platforms requires an integration tool such as the emerging Web Services (WSs) technology. The technology of web services has been proved of its effectiveness in the implementation of, usability and functionality services in the World Wide Web consortium [www.w3c.org] and the current most widely used business services delivered over the internet. Web Services technology simplifies the process of design, publication and further development of business services over the internet based on the wide number of easy to follow specifications and rules where even the existing services can use with easy techniques. These advantages of web services are exemplified in work "Introduction to Microsoft SQL Server and SQL Code Examples" by [1], [2], [3], [4], [5]. The technological advantages of web services are being seen in the building, application-based process driven websites for integrating into the websites framework applications that are different can be implemented in diversified environments and are independent. We can define a web service as an independent application which connects, communicates and processes the process with other web services using standard technologies of the Internet. Ex. SOAP (simple object access protocol) is used in an auction house application to place bids. A web service can help a composite business process such as "Travel Planner" to integrate its several different individual services such as, travel insurance, flight booking, hotel bookings, car rental and itinerary planning that can be sequentially or concurrently implemented.

In the numerous web services available today, a group of WSs may provide services of the same type that may group as a community. For selecting a web service from a community, the nonfunctional properties such as differences of pricing, are used. The non-functional properties termed as Quality of Service (QoS) characteristics are strongly suggested for selection of a web service. Web services-based technology is a promising technology for resolving system integrators issues of platform interoperability and compatibility. This web service selection process, however, does not use QoS factors in the selection process due to which the rate of acceptance has been low. The current implementation technologies of web services do not answer user questions such as, "what are the factors for determining if the web service will satisfy specific necessities of performance such as response time of 2 ms?" or "what are the reliability factors of the service, in implementing mission-critical systems?". As issues such as these remain unresolved, expectations of a business to look for a web service. That service meets its criteria in a UDDI (Universal Description, Discovery, and Integration) registry, [6], [7] are low, and there is no guarantee that a business without assurance of projected QoS delivery would implement the service. The multiplying of web services offering the business solution of integration for enterprise applications has made delivery of QoS the key concern for service providers. The challenges of web services implementation are ensuring QoS performance regarding the Web mechanisms that are erratic and dynamic as well and making network resources available to different applications contending for the WSs with different requirements. These challenges faced will make the service providers design better web services that achieve higher QoS, offering a competitive advantage over others. These processes must devise strategies to handle the complexities in determining and measuring the metrics of QoS. These QoS factors crucial for the success of distributed systems though have not been identified, properties such as availability, security, response time and throughput are recognized as key QoS characteristics.

1.2 QUALITY OF WS

The quality of web service characterized as the functional and non-functional associated factors quality level in terms performance, integrity, accessibility, reliability, interoperability, availability, and security. The highly complicated and multi-dimensional quality factors of web services require activity administration at various levels. However, the principal goal of the current strategies devised is meeting the user requirements regarding web service QoS, which decides the WS standard. Hence the qualitative aspects of web service have to concentrate on building the relationship between a prospective user and a specific service that eliminates the mindless search for an optimal solution.

The research on network infrastructure based QoS delivery has been extensive however the QoS of network alone will not ensure the complete QoS aspects wherein the server side will see a drop of high-priority traffic. For example, a Web site hosted by a system if does not have an integrated mechanism for offering complete QoS of services to users of priority, who may experience a drop-in traffic. The rules set for network QoS (performance isolation, service classification, and request admission, high resource utilization), though is specific to the Web server components, they form the basis for "quality of web-based services" (QoWS). The mechanisms and strategies by [8], implement QoS rules to host a Web site on a platform comprising of nodes of locally distributed servers (Web clusters) and the WSs are ascertained with experiments simulated based on SLAs (Service Level Agreements) and using users of different classes. With this study, the inference reached is that the rules of SLA will be met in varied load and system scenarios if they have dedicated system resources that dynamically change according to the set of users having the highest priority.

The primary criteria for finding a WS is to develop a framework that comprises of service provider QoS and user required QoS and merging of the two QoS will be the basis for exploring and determining a WS that meets both the specifications. The WS selection is more specific regarding the WSs provider where several QoS factors are considered, e.g., the QoS policy. E.g., an ideal policy would recommend for a process of a primary system involving a few WSs, however, would fail in a system contrasting it. In a system comparing the primary system, the SLAs created limit the QoS metrics of different types and can solve user requirements such as "what are the factors for determining if the web service will satisfy specific necessities of performance such as response time of 2 ms?" or "what are the reliability factors of a service in implementing mission-critical systems?". The process requirements necessitate the service providers to incorporate the system of regulating the user process management on the priority basis or implement various levels of QoS segregated on the cost basis to guarantee SLA for every user. The WSs effectiveness from the user point of view is based on a load of web services as well as other sources load experienced. The negotiation and monitoring of QoS require the user's inspecting the received service quality as well as the service providers tracking the user-generated load, to meet both the user and service provider's requirements [9]. An example is the experiments of [10] demonstrating QoS negotiation and adaptive WS selection.

In case of composite WSs, for each instance selecting the service application is a significant problem. The approach [11], a tuple (Fi, Qi, Ci) is allocated to every WS is, where Fi defines the functionality of the service's, Qi defines the attributes of the QoS, and Ci describes the cost. The attributes of QoS are metrics characterized effectively such that users and service are simultaneously clarified.

Also, a few metrics characterization based on a single value is not possible as they are based on the level of workload-intensity. The study by [11] for the flow of composite WSs is based on the selection a WS with five different variations from the same WSs collection. The factors involved concerning each scenario such as cost and the time, vary and these values differ when several service providers are involved. So, this requires selecting from the composite WSs the service applicable to the scenario. The study had considered only two attributes however the scenario would be justified if the number of attributes was considered. The strategy of using more number of attributes could handle by the theory of multi-attribute utility effectively. The WSs with similar characteristics of functionality are different in terms of functionality provisioning. For example, differences in the parameters of input and output requirements or the rules of implementation involving QoWS at varying levels, etc. as stated by [12]. This aspect is the basis for the techniques designed by [12] with a model that provides query services to various web services which are solved by merging all the obtained results with parameters of QoWS to fulfill the requirements of the user. This model enables the design and submission of the queries and subsequent query modification in the real web service process. The submission of queries of complicated nature to the WS is enabled for the first time by this model. The model furthermore in the context of semantic Web setup, emphasizes the significance of the role of WSs.

The future responsibilities of user's and service providers is going to be restrictive in terms of QoS specific to control and negotiation where evolution of complicated software systems will automatically increase the QoS necessity and the requirements to handle attacks, setbacks and for effectively achieving work-load balance in terms of variations automatically with inbuilt technology [11]. Hence applications that are QoS-aware with strong inbuilt components are the necessity. An application that is QoS-aware finds components that are QoS-aware which offer specific functionalities and is registered in a UDDI like service directory is given by [11]. Next, these are included after a successful QoS negotiation where the other components can avail the services of the component's that are QoS-aware in the QoS application. The execution of QoS negotiation is done with service requests in a sequence. The operations of four types that could execute by the component that are QoS-aware are, (a) services registration by means of a service registration module with a service directory. (b) QoS negotiation by means of a QoS request handler, a QoS negotiator, a QoS evaluator and a performance model solver that causes the QoS requests acceptance or rejection. (c) Provisioning of services for requests occurring concurrently by means of a service dispatcher maintains the QoS guarantees. (d) The QoS commitments made to other components are kept in a table format.

A QoS controller for e-commerce sites automatically establishes the best-fit configuration parameters conforming to the site's goals of QoS by [13] based on a routine algorithmic process for monitoring, the workload, and the site's various resources related performance data to maintain the QoS levels preferred.

A contemporary standard WSs set includes XML, SOAP, WSDL and UDDI for encoding, messaging, description, and discovery respectively that make the foundation of the expected complete WSs stack. The core-standard written in Business Process Execution Language (BPEL) is present in the top of the stack. The WSs standards of the existing stack, will improve with QoS standards quickly. The standard WS-Security is the first of these QoS standards to improve where important subsets of the security converted to standards' bodies. The standard WS- Trust for handling credentials of various secure domains with an XML syntax will shortly submit to OASIS. Immediately next, the standards body's official project is WS-Secure Conversation where the specification enables people in multiple message conversations eliminating the need for going with each new message to block one of the security checklist. The standard WS-Federation is assumed to reach the major standards phase within a year where multiple domain without a common single identity manager are provided security by the standard. The standard WS-Addressing, by the year ending may turn out as an official standard from the W3C. This standard makes possible asynchronous messaging and forms for each message a unique ID. The WS- Reliable Messaging, a key for QoS among others, will turn to be a ratified standard within a year. The IBM MQ model forms a message's place contained by a given specified sequence and offers to the involved parties fault notification with assured provisioning of one-time. The standard WS-Transactions would begin this year its standards life where it is used in the mainframes process for forming the required two-phase commits in the transactions process. The standard comprises of three components which are, WS-Coordination, WS-Atomic Transaction and WS-Business Activity. The component WS-Coordination in the deployment of the specifications of protocol will describe a framework. The component WS-Atomic Transaction will form transaction models of short period whereas WS-Business-Activity will handle transaction models of longer period of time. The WSs standards are based on the principle that as per requirement they may be mixed and matched. The WS-Policy will comprise of the precise requirements, certificates and encryption standards of the QoS for a specified service with a framework, extending the WSDL and implements automatic compliance parameters. Though in theory we would be able to get to a service and work devoid of human interaction however it expects from any day now onwards to be integrated into the standards body [14].

The model presented by us is a strategy that integrates the WSs discovery with the characteristics of the QoS. Here specifically we have designed a system that reveals information bits directly into WSDL [15] description or into a database of an intermediary [16]. In the process of determining the service is having QoS characteristics of maximum value, WSs discovery with similar functional factors is performed where the above-discussed factors of quality are extracted from the WSDL description and from the database which is then processed. This enables in achieving discovery of WS that involves quality as well as functional factors. We with the example of ticket issuing assess our designs with experimental simulation tests.

1.3 MOTIVATION

The work designed by us is based on the objective of improvising the WS discovery by designing WS discovery as per the quality aspects are given for preferred WSs [9]. Here we describe QoS as a set of non-functional factors that impact the WSs offered service quality. At present, a search for WSs via UDDI registry by a user is mostly based on functional factors where requirements of quality are not involved. So if WSs are having similar functional requirements however they may have factors of QoS that are different. For example, a customer who wants to use a service selling air tickets will also benefit if he is educated about the quality factors of every service he discovers involved in the business of selling air tickets. This will make the customer select or use the WS which offers QoS in terms of his necessities. Hence instead of the above factors, we present a WSs discovery which is based both on functional factors as well as requirements of quality.

1.4 PROBLEM DEFINITION

The authors Zheng et al., [17] depicted a novel goods ordering process (see Figure 1.1 follows) as an example to illustrate the issues of the QoS aware web service composition, which is the composition of seven web services with divergent providers, and each task has two services as possible solution (see Table 1.1 follows). The workflow of the illustrated example is as following (see Figure 1.1).

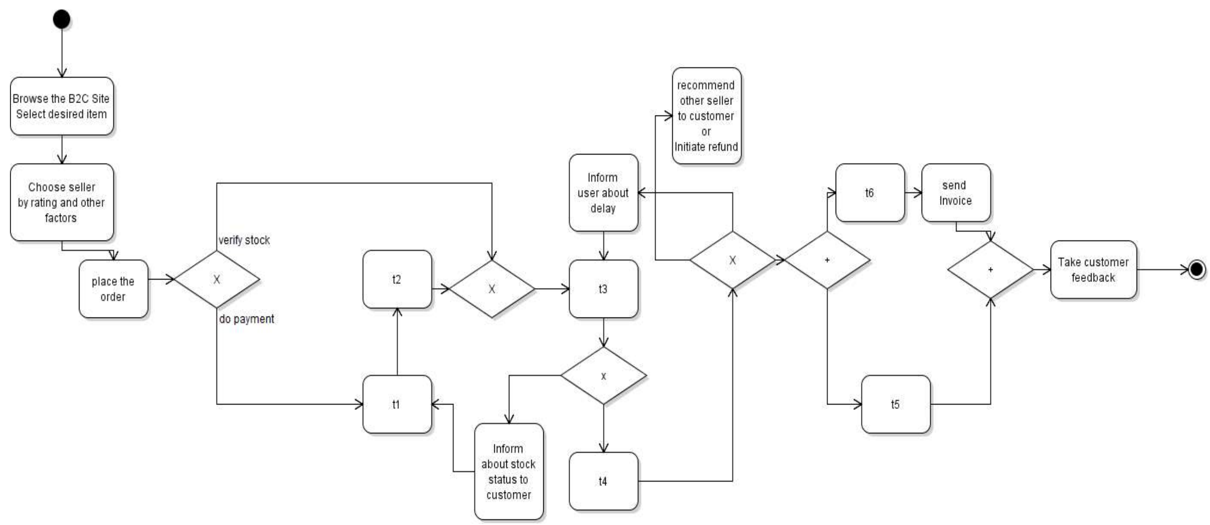

Figure 1.1: Workflow of the goods ordering process on e-commerce application

Table 1.1: Illustrative example of the available service providers for each task and their corresponding QoS attributes.

Provider	Tasks	Services	Respective Cost in cents	Respective response time in sec
A	t1,t2	$s1_A, s2_A$	1,2	0.2,0.2
B	t1,t2	$s1_s, s2_s$	1.5,5	0.1,0.15
C	t3,t4	$s3_c, s4_c$	1,2	0.2,0.2
D	t3,t4	$s3_D, s4_D$	1,5	0.4,0.25
E	t5	$s5_z$	1	0.2
F	t5	$s5_F$	2	0.2
G	t6	$s6_G$	1	0.2
H	t6	$s6_H$	2	0.2
I	t7	$s7_I$	1.5	0.1
J	t7	$s7_j$	5	0.15

Regarding placing a goods order, the customer initially registers the details of the goods, and then initiates the payment transaction or sometimes may verify the stock status and other particulars. If stock status verification is done and satisfied, then the payment transaction will be initiated, which is a credit card transaction. Once the card verified (task t1) then payment transaction (task t2) will be done successfully, that followed by the stock checking (t3) and reserving the registered goods for pickup (task t4). If stock not available, then the customer will update with the delay in goods delivery and tasks t3, and t4 of the composition waits some time and repeats. This wait, and repeat continue till max elapse time given to the seller. If he failed to deliver within the given time interval, then the customer should be updated with the same and recommend another seller if any. If no other seller is ready to dispatch that item or customer is not interested further, then the amount should refund. According to the depicted example, task t1 can accomplish by services s1A and S1B, t2 has the choice of services S2A and S2B. The services s1A and S2A owned by the provider A and other two services owned by provider B. According to the selection of composition context service of any provider can select to accomplish the tasks t1 and t2. If the composition context choice is exclusively the cost then the both services from the provider A will select, since the cost and is cheaper than the services owned by provider B. If response time is the choice of the composition context then the services S1B and S2B are preferable choice to accomplish tasks t1 and t2 respectively, this is since the response time of these two services is lesser than the services owned by provider A. If cost and response time both become choice of the composition context, then selecting services is becoming challenging task. Further, in the case of task t4, the service should invoke if ordered goods are in stock, else it waits till stock available. The stock verification is the task t3, and the selected service for the task should repeat the process of stock verification till stock available. The service that opted to accomplish task t4 may wait and repeat, which is based on the response of the service that opted for task t3. Henceforth, here in this case, rather QoS of the individual services, the connection between these two services plays the critical role. The Other composition context of the depicted goods order example is, accomplishing two tasks parallel. Once goods available in stock, the task t5 (pickup and delivery) and task t6 (sending the digitally signed invoice to the client through an email) should perform parallel. The QoS sensitivity involved in this is the failure of the service related to task t5 must not be tolerable against successful completion of the service related to task t6. Once the successful completion of the services related to tasks t5 and t6, the service in the schedule, which is to accomplish the task t7 (taking feedback from the customer) will initiate. Alongside these QoS challenges of composition context, few other nonfunctional QoS constraints such as max time limit of the process completion, max time limit of the composition completion will also play a vital role. The exploration of the service composition context of the depicted goods order example lets us accept the QoS aware service composition [18]. This service composition is challenging if most QoS metrics are considered and NP-hard [19], [20] if an increased number of services for each task with motivation gained from this model here we devised a novel soft computing strategy to select QoS aware service composition.

1.5 RESEARCH OBJECTIVES & CONTRIBUTIONS

- Regarding achieve the quality of service and secured activities from the web service compositions, they need to verify their impact towards fault proneness before deploying that service composition.
- Soft computing-based composition strategies are one of the significant dimensions of current research. But many of the existing benchmarking contributions limited to specific to the factors such as job completion time or consistency. In practice, achieving service quality is not optimal under limited QoS factors. The other constraint observed in the domain of soft computing strategies is computational complexity, which is found to be non-linear due to the substantially more evolutions owing to increment in available services.
- Services involved in composition practices of service-oriented architecture are needed to verify their impact towards fault proneness before deploying that service composition.

1.6 ORGANIZATION OF THE THESIS

The initial chapter of the thesis is to explore the domain of the research and significance of the Quality of Service-Centric web service composition. Further, the chapter-2 is the Contemporary Affirmation of the Recent Literature, which was done to explore the current benchmarking Service Composition Strategies and their limits. The chapter-3 that follows is elaborating our first contribution, which is a novel statistical approach to assess the service composition impact scale towards fault-proneness. Then the progress of the thesis was done through chapter-4 that explored a set of exploratory metrics, which enables to assess the services by multi-objective QoS factors. Chapter-5 that follows is examining a heuristic metric to evaluate the service composition impact scale for composition fault inclined (SCFI). The final chapter of the thesis is chapter-6 that explores the thesis summary and possible future research directions.

CHAPTER 2

2 REVIEW OF LITERATURE

2.1 OVERVIEW

Service compositions build new services by orchestrating a set of existing services. In the Internet of Services there may be many functional similar services, but with different Quality of Service (QoS). Thus, a significant research problem in service compositions is how to select the composition's composite services that the overall QoS of the composition is being maximized. This chapter summarizes, classifies and evaluates significant research efforts in this area and gives an overview of further open research questions.

2.2 PROLOGUE

The vision of the future internet is to transform the web of information to a web of services. The goal is to develop a so-called Internet of Services (IoS), where software modules of different complexity, analogous to the paradigm software-as-a-service, are provided on network servers and consumed via the internet.

In most cases, a single service is not sufficient to fulfill customers' complex requirements. In that case, service compositions are needed. A service composition orchestrates a set of services to one to solve a complex goal successfully [21]. The most well-known example of service composition is the travel planer that invokes a flight, hotel and car booking service in sequence.

Building a composition requires two steps [22]. At first, the functionalities required by the composite services (namely the tasks of the composition) and their interactions, the control and data flow, are identified.

Secondly, for each task, an appropriate implementation is selected and bound to the task. We call this set of service implementations execution plan of the composition.

As soon as there are at least two service implementations for one task, a selection of the numerous execution plans for the service composition has to make. In most cases, the execution plan selection is not done arbitrarily, but to maximize and/or limit some