RELIABILITY EVALUATION MODEL FOR COMPOSITE WEB SERVICES

N.Sasikaladevi[1] Dr.L.Arockiam[2]

[1]Lecturer, Department of Computer Applications, Bishop Heber College, Trichy, TN, India
sasikalade@yahoo.com
[2]Assosiate professor, Department of Computer Science, St.Joseph's College, Trichy, TN, India
larockiam@yahoo.co.in

ABSTRACT

Web services are the new innovation in this era. Because of the cross-platform and language independent nature of the web services, it is getting accepted by the various industries. Revolution in Service Oriented Architecture (SOA) has created numerous web service users. Different choices of services are available to meet the specific requirement of customers. At the same time, customer needs quality service. Selecting the web service which meets all the Quality of Service (QoS) requirement specified by the consumer in the Service Level Agreement is a tedious task today. One of the predominant QoS factor is reliability of the web service. Evaluation of reliability on web services and selecting the best one among the different choices are needed now. This paper focuses on the design of reliability evaluation framework for the composite web services.

KEYWORDS

Network Protocols, Wireless Network, Mobile Network, Virus, Worms &Trojon

1. INTRODUCTION

A Web Service is a software interface that describes a collection of operations that can be accessed over the network through standardized XML messaging. It uses protocols based on the XML language to describe an operation to execute or data to exchange with another Web service. It is a technology that allows applications to communicate with each other in a platform- and programming language-independent manner. Each business process consists of set of tasks. Every task is implemented by the set of services which is known as composite web services. Web service consumers depend on several related web service in order to complete their business process. Composite web services are evolving today.

With the widespread proliferation of Web services, quality of service (QoS) will become a significant factor in distinguishing the success of service providers. QoS determines the service usability and utility, both of which influence the popularity of the service. QoS will become an important selling and differentiating point of these services. QoS covers a whole range of techniques that match the needs of service requestors with those of the service provider's based on the network resources available.

Reliability, security, cost, and performance are criteria that are identified as relevant non-functional QoS requirements when selecting different web services. To evaluate the relative importance of these criteria a survey was conducted among several information technology architects in the Midwest [1]. The relative weights of these criteria were inferred through the analytic hierarchy process [2]. The results were consistent, and indicated that security and reliability were among the most important criteria that architects consider when evaluating web services. These results are depicted in Figure 1.

Fig 1. The relative importance of reliability factor on QoS metrics

This paper focuses on the reliability evaluation framework for the composite web services. Section 2 of this paper discuss on the related work that have been done on this area. Section 3 describes the new proposed model for the reliability evaluation. And section 4 gives the conclusion and further enhancements that can be made in this area.

2. RELATED WORK

Reliability is the ability of a system to keep operating over time. Xie [3] defines software reliability as "the probability that the software will be functioning without failures under a given environmental condition during a specified period of time". Web service reliability defined in two different aspects. There are message reliability and service reliability. Services are available over network with unreliable communication channel. Reliable delivery of message is a crucial task. Reliability of message is enforcing by two specifications, WS-Reliability and WS-ReliableMessaging. This paper focuses on the other reliability aspect called service reliability.

Hangjung Zo, and all [4] stated that Reliability for an individual web service is calculated as the number of failures per unit attempts, and is presented as a number between 0 and 1. The aggregated reliability of a business process that is composed from web services depends on the structure of the business process, the degree of independence between web services, and whether there are multiple web services (in the set of selected services) capable of supporting the task. Tasks in a business process can be related in a number of ways. The independence assumption permits simpler combining of reliability estimates, since joint probabilities are not involved. It can be argued that this assumption does not always hold, in that multiple web services from a vendor may prove to be unavailable if the vendor's site is down temporarily. They adopt the standard characterization of independence though. They assume overlapping functionality of the selected web services. Thus, a task may have alternate web services (in the selected set) capable of supporting it which will have an effect on the reliability associated with the task. They consider reliability at the task level, rather than the web service level. Redundancy of support for a task needs to be accounted for. They assume that failure at the task level constitutes failure of all alternate web services capable of supporting the task. Once again, independence between web services is assumed. The reliability of task *k* can then be computed as:

$$R_k = 1 - \prod_{j \in S} (1 - R_j)$$

where *S* represents the set of all web services that are selected to support the business process, and *Rj* represents the reliability associated with web service *j*. Only the failure rate of the web services is considered by Hangjung Zo, and all.

Abdelkarim Erradi and all [5] describe wsBus which provides various channels to access the registered Web services (each service is bound to one or more channels). Inbound messages are assessed on arrival through the channel to determine the destination service. Filters bound to the virtual endpoint, if any, intercept and manipulate both request and response messages (transform messages into new formats). The message is then passed through a reliability layer where it is checked for expiration, duplication, and ordering then it gets queued for processing. wsBus selects the appropriate service and dispatches the message to it and the response is passed back to the requester via the same path. The core idea of wsBus is to act as a bus which conveys SOAP messages from one end to another regardless the transport protocols (e.g.,HTTP, JMS) being used in either ends. wsBus sits between the transport and SOAP serialization/de-serialization modules, inspecting messages and relevant SOAP headers (e.g., SOAP Action header) as appropriate and providing requested added-value services as instructed by the configuration settings. A message store is used to enhance reliability by queuing and persisting messages. It can also serve as a system log for debugging and auditing purposes as well as for caching responses. wsBus is also equipped with a user friendly Management Portal to register and configure Web services, and to define access control policy for the registered services. wsBus Monitoring Console can be used to access QoS metrics and statistics gathered by the monitoring component during interactions as well as alerts raised in case of failures or violation of performance thresholds assigned to a registered service.

Luigi Cppolino and all [6] proposed an approach for reliability evaluation of workflow systems. In this approach a set of 20 basic workflow patterns is identified, which is suitable to describe virtually any *controlflow*. Starting from such patterns, they derive a set of new patterns – which they will refer to as reliability patterns - meaningful in the context of dependability. For each reliability pattern, they derive a rule which gives the reliability formula of the pattern. Since this reliability patterns are extracted from the workflow patterns identified, their formulas can be applied to a wider class of workflows. Another contribution of is a novel approach to the evaluation of the reliability of a single service, which explicitly takes into account the presence of infrastructure.

Vuong Xuan and all [7] suggested that indefinite QoS constraints are necessary to allow users flexibly specify their QoS demands in many cases of Web service applications. Fuzzy logic can be seen as a promising technique for representing such imprecise constraints. They presented an overview of prominent fuzzy based approaches for Web service discovery and selection. Various methods have been proposed for specifying fuzzy QoS constraints and for ranking Web services by basing on their fuzzy representation.

Gwyduk Yeom and all [8] introduced a Web Service QoS broker that acts as a UDDI service to the web service customers, but provides additional QoS information of web services. It measures the QoS of web services periodically and caches the information for fast services. Our current implementation uses response time, throughput, availability, and reliability as the QoS metrics. The measured data are statistically analyzed and saved in a database. When a user searches a web service, the Web services QoS broker system provides the analyzed information in numbers for software or in graphs for human.

3. RELIABILITY EVALUATION MODEL

Zhenua Liu and all [9] suggested a SOA(service oriented architecture) reliability evaluation model using two attribute: availability, which is the quality attribute of whether the web service is present or ready for immediate use, and accessibility, which is the quality attribute of service that represents the capable of serving a web service request. These two attributes are considered in this paper for the evaluation of reliability of the composite web services.

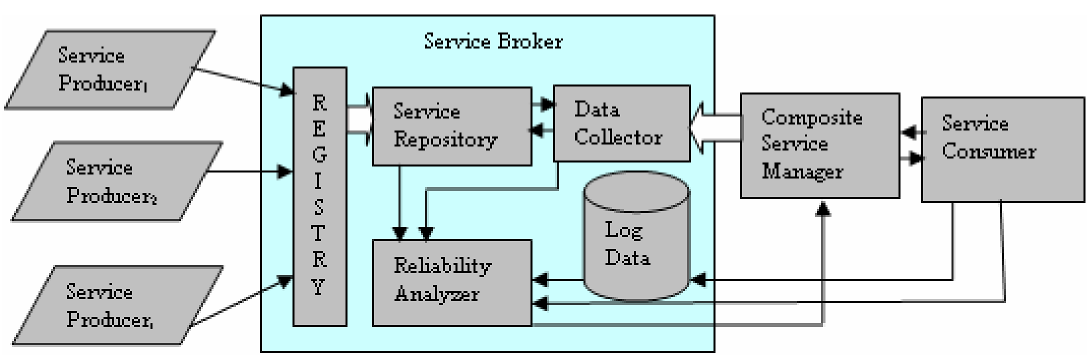

Fig 2. Framework model for Reliability Evaluation in Composite Web Service

Figure 2 describes the proposed framework model. This architecture is used to verify the reliability of each and every service involved in the composite web service. Initial web service structure involves three components: Service producer who is producing the service with the specific quality attributes, Service Consumers are going to use these services, and Service Broker is a middleware component in which service producers register their web service and service consumer discover the web service which meets their requirements.

SLA is an agreement between the service consumer and service producer in which quality requirement for the web services are specified clearly. Evaluation of this quality requirement to be done by the middleware brokers not to be done by the producer or consumer. So, in this architecture, reliability evaluation is done by the service broker behalf of the consumer. Let us consider the service consumer is in need to invoke a composite web service. Composite service manager is a component which is available in the service consumer receives this request from the consumer and identifies the set of services to be invoked.

Number of services involved in the composite web services is n. Composite service manager sends these services details to the data collector unit. Data collector component mine the service repository to verify whether these services are available or not. Service repository maintains the web service information specified by the service producers. For the evaluation of reliability, data are collected from producers. But this is inadequate to make the decision. Reliability evaluation can further enhanced with the use of feedback data from the consumer who already used the service. Log data maintains feedback information form the service consumer. The format of the log data is given below,

$$<S_i, V_i, X_i, t>$$

Where S_i is the service used by the consumer, V_i is the probability of availability, X_i is the probability of accessibility and t is the time at when the service is used by the consumer. Time value is used to identify the latest feed back value from the log file.

The core component of this model is the reliability analyzer which is responsible for making reliability evaluation. Reliability manager gets the services which are involved in the composite service from the data collector, and it receives the information about proxy for the each and every primary service. Number of proxies to be used for every primary service is m-1. It collects accessibility probability and availability probability of each and every primary service and its proxies from the service repository in which producer has given the probability values for these metrics and from the log files in which the most recently used consumer has given the probability values for these metrics. By using these values, reliability analyzer forms two matrixes: Availability matrix, Accessibility matrix. The Availability matrix is in the form:

$$\begin{pmatrix} v_{11} & v_{12} & \cdots & v_{1m} \\ v_{21} & v_{22} & \cdots & v_{2m} \\ \vdots & \vdots & & \vdots \\ v_{n1} & v_{n2} & \cdots & v_{nm} \end{pmatrix} \begin{matrix} V_1 \\ V_2 \\ \vdots \\ V_n \end{matrix}$$

where v_{ij} is the availability probability of the j^{th} proxy of i^{th} service. The Accessibility matrix is in the form:

$$\begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1m} \\ x_{21} & x_{22} & \cdots & x_{2m} \\ \vdots & \vdots & & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{nm} \end{pmatrix} \begin{matrix} X_1 \\ X_2 \\ \vdots \\ X_n \end{matrix}$$

where x_{ij} is the accessibility probability of the j^{th} proxy of i^{th} service.

Reliability analyzer then derives two vectors named V and X from these matrixes with maximum value on every row in each matrix.

$$(V_i, Xi) = \max ((v_{11}, x_{11}), (v_{12}, x_{12}) \ldots (v_{11}, x_{1m})) \text{ where } i=1 \text{ to } n$$

This vector pair is compared with the threshold value from the service consumer.

(Vi, Xi) >= Threshold, for all i in 1..n

If the values of the two vectors are greater less than or equal to the threshold value, then the proxy services with the maximum availability and accessibility probability values are selected and given to the composite manager. Then those services are invoked by the service consumer. If the values of the two vectors are less than the threshold value, then the proxies with maximum availability and accessibility probability values are maintained and next m-1 proxies are selected from the service repository and new vectors are designed and evaluation have to be repeated.

4. ILLUSTRATIVE RESULT

Consider the composite service C which is initiated by the consumer. Composite Service manager spilt the composite service C into set of individual services. Let us assume number of individual services are 4. Data collector component collects the information about these 5 services and proxies for each and every service. Let us assume number of proxies for each primary service is 4. If the number of available proxies is less than 4 then the null values are used to fill those entries. Reliability analyzer forms availability matrix by getting the data from the service repository. Then form another availability matrix by using the data from data collector. It finds out the average of these two matrixes.

$$\begin{pmatrix} 0.8 & 0.2 & 0.1 & 0.3 & 0.5 \\ 0.7 & 0.5 & 0.4 & 1.0 & 0.9 \\ 0.7 & 0.6 & 0.5 & 0.8 & 1.0 \\ 0.5 & 0.6 & 0.3 & 0.8 & 0.7 \end{pmatrix}$$

Then, the reliability analyzer forms the accessibility matrix by using the same procedure.

$$\begin{pmatrix} 0.7 & 0.2 & 0.1 & 0.4 & 0.5 \\ 0.6 & 0.5 & 0.4 & 0.9 & 0.9 \\ 0.8 & 0.6 & 0.5 & 0.8 & 1.0 \\ 0.8 & 0.5 & 0.3 & 0.8 & 0.7 \end{pmatrix}$$

Availability and accessibility vectors are formed by using maximum value on each row of these matrixes.

$$\begin{pmatrix} (0.8, 0.7) & (0.2, 0.2) & (0.1, 0.1) & (0.3, 0.4) & (0.5, 0.5) \\ (0.7, 0.6) & (0.5, 0.5) & (0.4, 0.4) & (1.0, 0.9) & (0.9, 0.9) \\ (0.7, 0.8) & (0.6, 0.6) & (0.5, 0.5) & (0.8, 0.8) & (1.0, 1.0) \\ (0.5, 0.8) & (0.6, 0.5) & (0.3, 0.3) & (0.8, 0.8) & (0.8, 0.8) \end{pmatrix} \begin{matrix} (V1, X1) = (0.8, 0.7) \\ (V2, X2) = (1.0, 0.9) \\ (V3, X3) = (1.0, 1.0) \\ (V4, X4) = (0.8, 0.8) \end{matrix}$$

Let us assume the threshold value from consumer is 0.7. Then evaluate whether V and X pair are having the value which is equal to or greater than 0.7. The suggested proxy services are,

S11, S24, S35, S44

The first proxy of the first service, fourth proxy of the second service, fifth proxy of the third service and fourth proxy of the fourth service are selected and its details are given to client.

4. CONCLUSION

Numerous web services are evolving today. Selecting the quality web service is a tedious process. One of the predominant QoS factor is reliability. This paper gives the simple framework model for the selection of reliable web service among the available proxy services. A reliability evaluation framework model is designed. Availability and accessibility parameters are used for the estimation. The probability values are received from the service producers and service consumers. Simple vector calculation is used to identify the reliable services.

REFERENCES

[1] Zo, H., *Supporting Intra- and Inter-Organizational Business Processes with Web Services*, Ph.D. Thesis, the University of Wisconsin-Milwaukee, August 2006.

[2] Saaty, T. L., *The Analytic Hierarchy Process*, McGraw-Hill, New York, NY, 1980.

[3] Xie, M., *Software Reliability Modeling*, World Scientific Publishing Co., Singapore, 1991.

[4] Hangjung Zo, Derek L. Nazareth, Hemant K.Jain, "Measuring Reliability of Applications Composed of Web Services", Proceedings of the 40th Hawaii International Conference on System Sciences-2007.

[5] Abdelkarim Erradi; Piyush Maheshwari, "A Broker-based Approach for Improving Web services Reliability", Proceedings of the IEEE International conference on Web Services (ICWS '05), 2005.

[6] Luigi Coppolino; Luigi Romano; Nicola Mazzocca; Sergio Salvi, "Web Services Workflow Reliability Estimation Through Reliability Patterns", IEEE Computer Society.

[7] Vuong Xuan TRAN; Hidekazu TSUJI, "QoS based Ranking for Web Service: Fuzzy Approaches, Proceesings of 4th International conferences on Next Generation Web Services Practices, 2008 Page(s):77-82.

[8] Gwyduk Yeom; Dugki Min, "Design and Implementation of Web services QoS Broker", Proceedings of the International Conference on Next Generation Web Services Practices (NWeSP'05), 2005, IEEE Computer Society.

[9] Zhenyu Liu; Ning Gu; Genxing Yang; "A Reliability Evaluation Framework on Service Oriented Architecture", Pervasive Computing and Applications, 2007. ICPCA 2007. 2nd International Conference on 26-27 July Page(s): 466-471.

[10] Jiang Ma; Hao-peng Chen, "A Reliability Evaluation Framework on Composite Web Service", Proceedings of IEEE International Symposium on Service Oriented System Engineering, IEEE Computer Society, 2008, Page(s): 123-128.

[11] Jia Zhang; Liang-Jie Zhang, "Criteria Analysis and Validation of the Reliability of Web Services-oriented Systems", Proceedings of the IEEE International Conference on Web Services (ICWS'05), IEEE Computer Society, 2005.