Volume 4, Issue 4, April 2014 ISSN: 2277 128X

International Journal of Advanced Research in Computer Science and Software Engineering

Research Paper

Available online at: www.ijarcsse.com

A Survey Report on Reliability Models and Frameworks in SOA

Mousumi Rout, Prachet Bhuyan

School of Computer Engineering,

KIIT University, Bhubaneswar, India

Abstract- SOA is a popular paradigm for development of distributed systems.SOA allows interaction of various application based on different platform, implementation language and locations. Generic services are the core of SOA. Revolution in SOA has created growing number of service users, hence there is a need of high quality services. One of the predominant quality is Reliability. Reliability can be described as probability that the system should work properly as per the requirements without any failure during a given time duration under stated conditions. We have analysed existing frameworks , models or methodologies from the viewpoint of reliability in SOA and overview of those research works is given. At the end of this paper we have discussed some reliability challenges still exist in this area.

Keywords-SOA Reliability, Reliability Evaluation, Service Reliability, Reliability prediction, Fault Tolerance Mechanism

I. INTRODUCTION

Service Oriented Architecture is an architectural style for creating an Enterprise IT architecture that exploits the principles of service orientation to achieve a tighter relationship between the business and the information systems which support the business . The Primary goal of SOA is to align business world with world of Information Technology(IT) in a way that makes both more effective. SOA treats business components as well IT infrastructure as components that can be reused and recombined to address changing business priorities[12]. SOA focuses on developing reusable IT components that contains business functionality and SOA also focuses on designing the business itself as a set of reusable business functions that can be automated partially or completely by IT. A client application simply consumes the service through a well defined interface and the service providers takes care of the implementation. Web Services technology is the most common implementation of SOA. Security, reliability, cost, and performance and other quality metrics or non-functional criteria when selecting services. One of the predominant quality is reliability, IEEE definition of reliability is The probability that a software will not cause failure of a system for a specified time under specified conditions. Reliability specifically refers to the continuity of the service delivered by a system. In this respect, reliability can be described (i) the probability that the system performs its required functions under stated conditions for a specified period of time and (ii) the probability that the system successfully completes its task when it is invoked (also known as reliability on demand).There is a high need of reliable system because unreliable system not only shorten the age of the system but also it cause loss in terms of time and money.

A. Some Reliability Challenges in SOA

Examining reliability should consider reliability of each component involved in distributed application and factors influencing reliability. Some of the important factors like uncertainty in communication link, hardware or other resource failures, issues with changing management or operational environment should not considered constant while analyzing reliability. While detecting faults and evaluating reliability all the factors should be taken into consideration. Fault detection techniques should able to detect broad range of faults. Failure detection methods should be suitable for large scale distributed environment like SOA environment also should be suitable for heterogeneous, dynamic environment .

Resource sharing in large scale systems leads to failures like timeout failures, blocking failures, program failures which must be analyzed during building reliable systems.

Failure data are collected for evaluation of reliability. These data collected through in house testing should not be compared with failures which can occur under actual operational environment. Failure data must be collected during real time running application.

In Service-Oriented Architecture based applications services can be owned and hosted by different organizations. Developing distributed systems by integrating these services and predicting and analyzing reliability of such systems is a challenging task. Some models and techniques have been proposed to assess evaluation or predict reliability in SOA but some reliability issues still exists. This paper focused on systematic survey on some reliability models and frameworks used for reliability measurement, prediction at early stage, fault tolerance or reliability enhancement in SOA.

The rest of the paper is arranged as follows: Section II describes the related work. Section III we have presented a comparative study on models and frameworks for in SOA. In Section IV describes some discussion regarding our literature survey and some issues still present in the above area. Section V presents the conclusion and future work.

II. RELATED WORKS

Delac et al. presented a reliability model for SOA in which they estimated reliability of atomic service by a testing framework .They also estimated reliability of composite service using a directed acyclic graph. Authors also provided a simple weak point detection method in the composition reliability model .The method use a simple strategy to locate weak points in a composition. According to the authors weak points can be strengthen by using a better fault tolerance methods. Authors suggested that this approach can be further improved by using an efficient weak point detection method and clarifying the service discovery and service selection methods . The accuracy of the model presented in this paper is influenced by the accuracy of reliability estimation for atomic services and the accuracy of his estimation depends on the frequency of conducted tests, as well as the test coverage for the faults related to improper service. Introducing weight factors representing the influence of a service on the composition can further improve the model. For each service S1…Sn from the service repository a corresponding monitoring process M1…Mn is maintained. Each monitoring process is responsible to periodically test the service by invoking all its methods. In the proposed framework, reliability of each method is stored separately. One of the approaches to predict reliability is to maintain the history of the method's invocation attempts. The reliability can then be estimated by dividing the total number of fault occurrences by the number of invocation attempts. In order to get a precise reliability measurement, the service tests should be relatively frequent, but, on the other hand, they should not have a severe impact on the service's performance itself. In addition, the tests should be run for some time to get a wider usage history[1].

Zhang et al. proposed a model to estimate the reliability of the Web services. This testing framework assess reliability for Web services and integrate into the Web service architecture . The proposed reliability model evaluates both atomic web service reliability as well as the overall composite service reliability. A voting method is used in this model instead of monitoring system. Each atomic service is assigned with a number of services with equivalent functionality. The faults are detected by applying the majority vote principle, i.e. the fault is detected if consensus cannot be reached. The presented model calculates the reliability of atomic services in regular time intervals when the votes come after interval.. A clear advantage of this approach is the ability to measure the reliability during system's operation. In this approach various replicas with same functionality of a service must be there, which increases resource demand. The reliability values of atomic services are used to compute the service composition reliability. Authors uses a scenario based approach where scenario is weighted by its execution rate and can consist of operations executed in a sequence by atomic services.[2]

Liu et al. proposed an approach which calculates the overall quality of service of individual services in a service composition. The estimated quality serves as guidance for the developers to detect weak points in their compositions. For each service a quality matrix is maintained which can hold multiple quality parameters. Authors suggested 2 reliability metrics those are availability and accessibility and monitoring of these metrics. Also, an involvement matrix is used to know how the functionalities of a particular service influences in a service composition. By using these two matrices weight factors for each service can be calculated. The weight factors are multiplied by service quality parameters to get the overall quality of service. The authors propose a monitoring environment in which the services are accessed through a unified mediator. The mediator performs quality measurements and stores the results in the repository. The analyzer then extracts the measurements from the repository and performs the reliability estimations[3].

Jiang et al. proposed a web service reliability model for atomic web service without structural information and the composite web service consist of atomic web services and its redundant services. Authors proposed a framework based on client feedback to gather attributes to service registry used for reliability evaluation. The paper represented a reliability model of web service and its redundant services to evaluate composite web service reliability. The introduced evaluation framework which is based on collecting consumer feedback for evaluation framework. As for any attributes of quality, this framework can configure more new attribute along with the previous one through the suitable model [4].

Wang et al. proposed a hierarchical reliability model, in which the system reliability is calculated according to the layered superimposition of the reliabilities of data, services, fault-tolerant mechanism and service composition logic. At the composition layer, a Discrete Time Markov Chain model is created for analysing system reliability based on the reliability of the constituent services and their execution rate which are decided by the composite control structure and the operation scenarios. The DTMC state transition diagram can be automatically generated by analysing and transforming the layered superimposition of the application following the transformation rules. At the basic-service layer, the service reliability is considered in the context of data reliability and service pools with backup alternatives. The hierarchical modelling framework enables change adaptation at various levels. In addition, the models can be continuously learned and dynamically adjusted by profiling on the runtime monitoring log files. The reliability model is adaptive to the changes in the service composition and system configuration dynamically[5].

A Hybrid Reliability Model based on log analyzer is designed by Maruthurkarasi et al. to evaluate the Reliability of Composite Web Services. Based on Dependability, Atomic Web Services are composed with a Central Co-ordination Function (Broker). Real Time Server Log Files are fed as input to the system. Log Analyzer reads the log entries and separates the individual response of the server along with the time stamps. Base on the Frequency of the service response are classified and the Error Rate is calculated by the difference in the Uptime and Downtime Stamps. The Broker designs and decides the acceptance of the service based on Error Rate (MTBF, MTTF, MTTR) and Fault Tolerance. As, the Error Rate and Service Reliability are inversely proportional, the server with low error rate provides high reliability. Their Experimental Results with their groupings prove that the reliability can be evaluated using the Web Log File analysis. By analysing the unique problems and information sources for the web environment, an approach for identifying and characterizing web errors and for assessing and improving web site reliability based on information extracted from

existing web logs is developed. The results demonstrated that the error distribution across different error types and sources is highly uneven. In addition, missing le distributions, workload distribution, as well as reliability distribution for individual types of requested are all quite uneven. This approach can help web site owners to prioritize their web site maintenance and quality assurance effort and to guide further analyses, such as root cause analysis, to identify problem causes and perform preventive and corrective actions[6] .

Xie et al. proposed a model based on birth death process. They used fault tolerance by redundancy. According to that model some set of services and backup services are maintained. A web service manager redirects all the request for services. When a service is unavailable or in case of service failure the manager directs the request to similar service and notifies UDDI for rectification of the failed service. Authors calculated reliability for the services and also measured reliability when the fault tolerance technique is applied.[7]

WS DREAM is a user centric approach where those who carry out testing share their result through a centralised server. Reliability is calculated using overall data.[8]. Passive and active replication technique dynamic service replacement techniques are used for failure recovery.[11],[10]. Zheng et al. predicted reliability during design phase using past failure data of similar services used before.[9].

III. COMPARISON TABLE

After reviewing some of the existing works in the interested area we have given a comparative survey on reliability models and frameworks in SOA as follows:

TABLE I
COMPARATIVE SURVEY ON RELIABILITY MODELS AND FRAMEWORKS IN SOA

Sr No	Paper Title/Model	Aim of the paper	Proposal	Remarks
1.	SORM(2004)	Evaluates Reliability of atomic service as well as composite service.	Group testing and majority voting principles are used to evaluate service reliability. Composite service reliability is evaluated using architecture based model and from reliabilities of atomic services and operational environment.	There can be situations when there is no alternate service available. Here SORM may not work well.
2.	Reliability Evaluation Framework on SOA(2007)	Compound framework which calculates some QoS attributes.	Running status and possible status of each service node is compared using matrix. A monitoring system is used where a mediator is used which accesses services. The mediator also performs quality measurement and stores the results in the repository .Then one component analyzer extracts data for repository and performs reliability calculation.	The framework could be more flexible. When there is change in requirement and new functions are added or deleted ,or modified this framework would increase cost.
3.	WS DREAM(2008)	A user centric reliability assessment mechanism for web service.	This model is used to assess reliability in SOA based application.	Network connectivity among different components, server problems were not

			In this approach all users carry out testing under coordination of centralized server. Then users share their results on a central server which makes assessment of web services much easier.	considered.
4.	Designing Fault Tolerant Web Services using BPEL(2008)	Failure recovery technique.	Used both active and passive replication technology. They used the fault tolerant technique in web service orchestration. This model in basically based on 3 principles composition and invocation of service, failure detection, fault tolerance .	This model provides backup services but maintaining service state is still an open issue.
5.	A Reliability Evaluation Framework on Composite Web Service(2008)	Evaluates atomic and composite service reliability .	This model is based on client feedback .The UDDI holds the feedback information from the clients when the client invokes and consumes service ,which is used for evaluating reliability.	The challenge of this model is the trustworthiness and accuracy of the information they provide through feedbacks.
6.	Collaborative Reliability Prediction of Service Oriented systems(2010)	Predicts reliability at design phase in SOA	This model helps in reducing reengineering cost .Users share failure data. And this approach predicts reliability for similar set of users and services based on past data.	Considered similar service but did not consider similar infrastructure which is an important factor in determining reliability .
7.	Dynamic Service Replacement to Improve Composite Service Reliability.(2011)	Reliability enhancement by replacing service in composition dynamically.	Authors stated 3 types of failures during WSBPEL composition also used aspect oriented programming. Failure service is replaces by alternate service dynamically.	Can be improved by proper composition and service selection during replacement dynamically.
8.	Reliability Model based on Birth-Death Process(2011)	Reliability measurement and reliability improvement.	Extended SOA conceptual model used redundancy technique. In this	This model did not consider reliability of composite services and also some other factors are

			model service manager associated with each service look up services when a request comes. When the service is unavailable is redirects to backup services. Also Reliability is measured before and after rectification of failure.	not considered.
9.	Reliability Modelling in SOA(2012)	Calculates reliability of atomic and composite web service	This model calculates atomic service is reliability by utilizing a testing framework .They calculate reliability of composite service using a directed acyclic graph. When fault is detected model is improved by using fault tolerance methods.	Authors provided a simple method for detecting weak points in the composition reliability model .The method applies a simple strategy to locate weak points in a composition. This approach can be improved by using an efficient weak point detection method and clarifying the service discovery and service selection methods when fault tolerance approach is applied.

IV. DISCUSSION

In this paper we have analyzed some existing reliability models and frameworks in SOA which contribute towards predicting, evaluation of reliability and fault tolerance techniques. The above analysis revealed that reliability of a SOA based system can be ensured by :

- Proper testing of errors during development
- Predicting reliability at early stage(ensuring reliability at design phase)
- Accuracy in measurement of reliability
- Providing efficient fault tolerance techniques when failure occurs.

Some of the open issues still survive are

Maintaining service state e.g when a failure occurs between a transaction maintaining the state before and after transaction .

Issues like database connection failure , network issues, internet host problem, application server problem should be taken into account in reliability models.

Necessity of better service selection criteria and efficient service publication which helps in knowing QoS information about a service.

Fault tolerance techniques should consider performance overheads.

There is a need of clear monitoring system which can detect broad range of faults quickly without causing any performance issues and monitoring of parameters required for evaluation of reliability.

All reliability influencing factors should be considered in reliability models.

V. CONCLUSIONS

Aim of this survey was to analyze some of the existing works related to reliability in SOA and highlighting some of the open issues still survive. We have mentioned the issues where improvements to existing works can be done. Our future plan is to develop a monitoring system which can detect broad range of faults and parameters required for measuring reliability efficiently.

REFERENCES

[1] G. Delac, M. Silic and S. Srbljic "Reliability Modeling for SOA Systems "*IEEE Transactions on Software Engineering* (May 2012).

[2] D. Zhang et al. "A Software Reliability Model for Web Services", *The 8th IASTED International Conference on Software Engineering and Applications*, 2004, pp. 144-149.
[3] Zhenyu Liu, Ning Gu and Genxing Yang , "A Reliability Evaluation Framework on Service Oriented Architecture", *2nd International Conference on Pervasive Computing and Applications* (IPCA 2007), October 2007, pp. 466-471.
[4] Jiang Ma, Hao-peng Chen , " A Reliability Evaluation Framework on Composite Web Service" *IEEE International Symposium on Service-Oriented Engineering 2008* p123-128.
[5] Lijun Wang, Xiaoying Bai, Lizhu Zhou , Yinong Chen "A Hierarchical Reliability Model of Service-Based Software System". *IEEE Transactions on Software Engineering* 34 (2009), 199-208.
[6] Rohini, C.Jayaprakash, Ganesh, R. B., and .Maruthurkarasi, P. "Reliability Enhancement from http Log Files In Composite Web Services", *International Journal of Engineering and Advanced Technology (IJEAT*) vol 2- 4 (2013), 464-468.
[7] Chunli Xie, Bixin Li, Xifeng Wang "A Web Service Reliability Model Based on Birth-Death process" Tech. Rep.
[8] Zibin Zheng, M.R. Lyu, "WS-DREAM: A distributed reliability assessment mechanism for Web services" , in*: IEEE International Conference on Dependable Systems and Networks with FTCS and DCC, DSN* 2008, 24–27 June 2008, pp. 392–397.
[9] Z.Zibin Zheng, Michael R.," Collaborative reliability prediction of service-oriented systems", in: *Proceedings of the 32nd ACM/IEEE International Conference on Software Engineering*, vol. 1, ACM, Cape Town, South Africa, 2010.
[10] Jong-Phil Kim, Jang-Eui Hong, "Dynamic Service Replacement to Improve Composite Service Reliability" , *Fifth International Conference on Secure Software Integration and Reliability Improvement* 2011,p-183-188.
[11] Lau et al., "Designing fault tolerant web services using BPEL", in: *Seventh IEEE/ACIS International Conference on Computer and Information Science, ICIS* 08, 14–16 May 2008, pp. 618–623.
[12] Article Available : http://www.teknoglobal.com/index.php?option=com_content&view=article&id=104;Itemid=239.

