December 2015

Vol 2, No 12 (2015)

Sai Om Journal of Science, Engineering & Technology: A Peer Reviewed National Journal (Online ISSN 2347-7547)

Table of Contents

Articles

ISSN: 2347-7547

Published by: Sai Om Publications

Sai Om Journal of Science, Engineering & Technology

A Peer Reviewed National Journal

EVALUATE THE RELIABILITY OF WEB SERVICE MESSAGE EXCHANGE USING AGENTS IN SERVICE ORIENTED ARCHITECTURE

C. Jayaprakash[1]
Research Scholar Sathyabama University, Associate Professor in Department of Information Technology, KLN College of Information Technology, Sivagangai, India
Email: jpeverjp@gmail.com

Dr. V. Maheswari[2]
Professor and Head of Computer Applications, Sathyabama University, Chennai, India
Email: maheswarikarthikeyan@hotmail.com

ABSTRACT

Web Service is a very powerful tool that has greatly enhanced the efficiency of communication among businesses. Message exchange in Web services can be inherently unreliable. To address this challenge, in this paper we propose an architecture for reliability evaluation. Various Scenarios along with reliability attributes of the proposed architecture are evaluated using ATAM (Architecture Tradeoff Analysis method). In addition, a Dynamic Evaluation of web service reliability using agents for the proposed architecture is done using RWS Evaluation Method.

Keywords: Web Service; ATAM; RWS; Dynamic Evaluation

INTRODUCTION

Web Service providers provide web services with the same functionality, but these services can be distinguish from each other by non-functional aspects such as reliability In the context of Web services, we can address the issues of reliability at several distinct levels: the reliable and predictable delivery of infrastructure services, such as message transport and service discovery, of reliable and predictable interactions between services, and of the reliable and predictable behavior of individual requester and provider agents. The ultimate vision is to enable a client to automatically identify service from a pool of dynamically discovered services and invoke it without having any prior knowledge about the service provider.

Reliability, security, cost, and performance are criteria that are identified as relevant requirements when selecting different web services.[1]Among these criteria, reliability evaluation is most important functional requirement which we have discussed in this paper.

OBJECTIVES

1. To evaluate reliability of web services with attributes using ATAM (Architecture Tradeoff Analysis method).
2. To make Dynamic Evaluation of web service reliability using agents in RWS Evaluation Method.
3. To provide ranking for each service.

LITERARY SURVEY

Web services used primarily as a means for businesses to communicate with each other and with clients, Web services allow organizations to communicate data without intimate knowledge of each other's IT systems behind the firewall. Web services instead share business logic, data and processes through a programmatic interface across a network.

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-Reliable Messaging. This paper focuses on the other reliability aspect called service reliability. Abdel karim Erradi and all [4] describe WS-Bus 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. WS-Bus 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 WS-Bus 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.

Luigi Cppolino and all [5] 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 control flow. 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. Thirumaran et al. [2] provided a sophisticated architecture for quality driven web service evaluation. According to [2] agents are used to evaluate the QWS parameters. They also discussed about the quality attributes with organized set of design related questions which helps an evaluator to analyze the ability of the architecture to meet quality requirements, and provides a brief sample evaluation. ATAM method of software architecture evaluation is used to evaluate the proposed model of [2]. The assessment justifies the proposal in terms of the performance attributes such as reliability, availability, modifiability, security and interoperability etc.

In this paper we have designed an architecture based on ATAM method. A Dynamic Evaluation of web service reliability using agents for the proposed architecture is done using RWS Evaluation Method.

Proposed System

The proposed architecture named REWS Architecture is shown in figure 1. The UDDI Registries contains information published by businesses about the various services they provide. User gives his service request to request handler. Service Registration and Profile Handler handle the request using UDDI – WSDL. REWS Parameter Analyzer is an important part of the architecture where analysis of various parameters like guaranteed delivery, ordering the service, fault tolerance and duplicate elimination were done. Criteria Value injector generates various scenarios for the REWS Parameters. The evaluation is done making use of the information from server log files, certifier, service log file, service usage pattern, service profile and feedback. The ranking of the web services is done with the

help of RWS evaluation agent and knowledge base and a list of web services meeting the client's criteria is provided among which the client can make a choice of the web services. This prioritized list is stored in the service pattern database for future use if same kind of request is made.

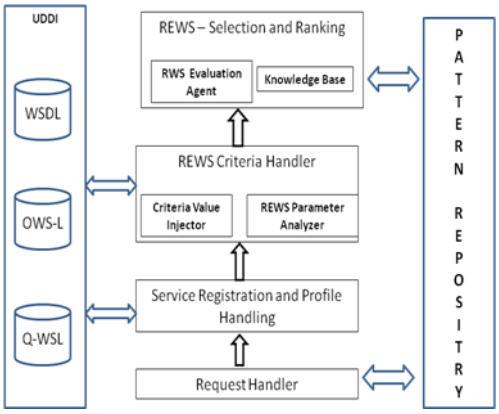


Figure 1. REWS Architecture

REWS Engine shown in figure 2 evaluates various reliability parameters and after categorizing them submits to respective analyzer which in turn performs analysis of the parameters. Agent Monitor evaluates and stores RWS parameters into knowledge base. Agent Rule Engine is then used to select a reliable service.

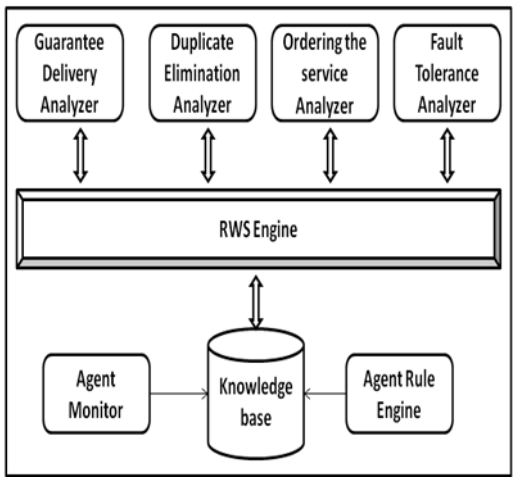


Figure 2. Internal Structure of REWS - Evaluation Agent

Evaluation of Proposed Architecture

Evaluation of the proposed architecture is based on ATAM (Architecture Tradeoff Analysis method) [2][6][7][8]. All the scenarios corresponding to each application of the service mining are listed and evaluated based on reliability attributes. The figure 3 shows the reliability tree.

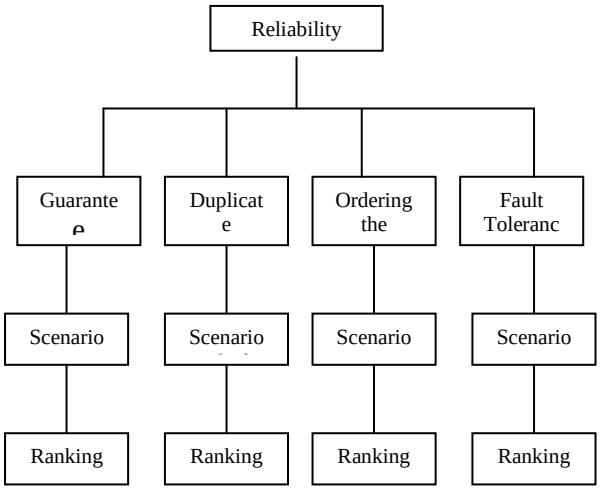


Figure 3. Reliability Tree

The reliability tree provides a mapping between the reliability attributes that the architecture to meet discussed in business driver to the corresponding scenarios. In this tree, root node is "reliability" and the second level node are various reliability attributes of architecture and third level represents various scenarios for each attributes; fourth level represents pair of ranking to represent the priorities of nodes in leaves. The scenarios are prioritized relative to each other using ranking pairs of (High, Medium and Low). It would be (H, H), (H, M), (H, L)etc..The first letter denotes the degree of importance to system and second letter denotes degree of difficulty in achieving it. The scenario prioritization is included in the Table 1.

Table 1. Scenario Prioritization

Scenario No.	Scenarios	Reliability Attributes
Scenario1	Receiving Acknowledgement	Guarantee Delivery
Scenario2	Match the no of acknowledgement received with no of messages sent	Guarantee Delivery
Scenario3	Uniqueness of the messages	Duplicate Elimination
Scenario4	Late Delivery of Messages	Ordering the service
Scenario5	Late Delivery of Acknowledgement	Ordering the service
Scenario6	Availability of Sender or receiver	Fault Tolerance
Scenario7	Receive message in the original Format	Fault Tolerance

Architecture Analysis

Architecture analysis is to reveal enough information about the proposed architecture to identify its risks, non-risks, tradeoffs, and sensitive points in the design phase itself rather than later phase. This method is not meant with precise and detailed evaluation of architecture reliability attributes with its numerical value. In the evaluation phase, we identified risks associated with architectural decisions and their effects on reliability attributes. Table 2 shows with defined scenarios and their risks, tradeoff, scenario prioritization.

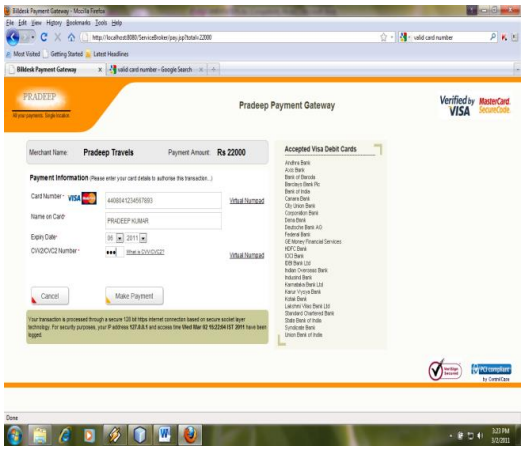
Table 2. Analysis of scenarios in Architecture

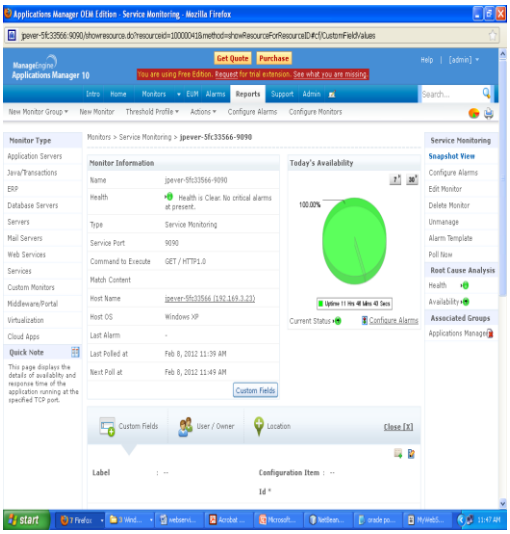
Reliability Attributes	Scenarios	Goals	Risks	Tradeoffs	Priority
Guarantee Delivery	Scenario1	Confirmation of Delivery	Loss of Acknowledgment	Network Error	<H,H>
Guarantee Delivery	Scenario2	Assure Complete Message Delivery	Loss of Some Acknowledgment	Network Traffic	<H,M>

Duplicate Elimination	Scenario3	Avoid Repetition	Source may resend the same message	Loss of Acknowledgment	<H,M>
Ordering the service	Scenario4	Message - Reach in time	Source may resend the same message	Loss of Acknowledgment	<H,H>
Ordering the service	Scenario5	Acknowledgment -Reach in time	Source may resend the same message	Loss of Acknowledgment	<H,H>
Fault Tolerance	Scenario6	Mutual Availability	Message Lost	Server or client may not be active	<H,M>
Fault Tolerance	Scenario7	Receive message in the original Format	Format May be changed	Due to Encryption and Decryption	<H,M>

Implementation

We implemented for Tourism application using net beans with GlassfishServerV2. Here we have created web service for Boarding and Lodging, another web service for Travel plan and one more web service for Bill settlement. Application maintains the customer profile, transport facilities information, Boarding and lodging related information. The Qos such as computability, traceability, accessibility is evaluated using Application Manager 7. After the evaluation is completed a generated based on the evaluation using Application Manager 7.





Reliability of Web Service (RWS) Evaluation

Calculation of Performance for Guarantee Delivery

For zero millisecond (ms) = 100 points for every 10ms(x) = 1 point deduction

Gross points (GP) = (100-((response time)/x))*weight)

(Take weight as 5 for response time and 4 for latency)

Capacity (ms)=No of ACK received / time taken(ms)

Calculation of Fault Tolerance

Accuracy GP = (100 –((No. of Format Change)*10)*weight

(100 points for original content, deduct 10 points for each format change)

(Here weight = 5 when received successfully otherwise zero)

Availability GP = (1-(Down Time/Measurement time))*100% *weight

Accessibility GP= (Number of response messages/ Number of request messages)*100 % * weight

Success GP = (Number of response messages/Number of request messages)*100%*weight

Usability = Average percentage of Accessibility and Success) * weight.

CONCLUSIONS

In this paper we proposed a architecture for reliability evaluation. Various Scenarios along with reliability attributes of the proposed architecture are evaluated using ATAM (Architecture Tradeoff Analysis method) which allows identifying the duplicates, fault tolerance and priority of each scenario. In addition, a Dynamic Evaluation of web service reliability using agents for the proposed architecture is done using RWS Evaluation Method.

REFERENCES

1. Abdel karim Erradi; Piyush Maheshwari, “A Broker-based Approach for Improving Web services Reliability”, Proceedings of the IEEE International conference on Web Services (ICWS ’05), 2005.
2. Bin Li; Xiao-Yan Tang; Jian Lv, 2005 .“The Research and Implementation of Services Discovery Agent in Web Services Composition Framework ” Proceedings of 2005 International Conference on Machine Learning and Cybernetics, Volume 1, Issue , 18-21 Page(s): 78 – 84, Aug. 2005.

3. Colin Atkinson and Philipp Bostan,"A Practical Approach to Web Service Discovery and Retrieval" IEEE International Conference on Web Services, ICWS 2007.
4. Chhabra, Manish Lu, Hongen La Trobe University, Australia, 2007."Towards Agent Based Web Service", International Conference on Computer and Information Science, On page(s): 93-99.
5. Janette Hicks, Madhusudhan Govindaraju, Weiyi Meng, "Search Algorithms for Discovery of Web Services" IEEE International Conference on Web Services, ICWS 2007.
6. Luigi Coppolino; Luigi Romano; Nicola Mazzocca; Sergio Salvi, "Web Services Workflow Reliability Estimation Through Reliability Patterns", IEEE Computer Society.
7. M. Thirumaran et al., Architecture for evaluating Web Service QOS Parameters using Agents, International Journal of Computer Applications (0975-8887), Vol 10, No.4, November 2010.
8. N. Sasikaladevi, Dr. L. Arockiam, "Reliability Evaluation Model for Composite Web Services", International Journal of Web and Semantic Technology (IJWEST), Vol1, No.2, April, 2010.
9. Qianhui Liang, Jen-Yao Chung, Steven Miller, Ouyang Yang, 2006." Service Pattern Discovery of Web Service Mining in Web Service Registry-Repository", IEEE International Conference on E-Business Engineering, Pages: 286-293, 2006.
10. Xie, M., Software Reliability Modeling, World Scientific Publishing Co., Singapore, 1991.
11. ZHANG Changyou1 and ZHU Dongfeng, "A Web Service Discovery Mechanism Based on Immune Communication" International Conference on Convergence Information Technology, 2007.