2011 International Conference on Network and Electronics Engineering
IPCSIT vol.11 (2011) © (2011) IACSIT Press, Singapore

Web Services Metrics: A Survey and A Classification

Mohamad Ibrahim Ladan, Ph.D.

Computer Science Department, Haigazian University
Beirut – LEBANON
mladan@haigazian.edu.lb

Abstract. Web Services have emerged as a new Web-based technology paradigm for exchanging information on the Internet using platform-neutral standards, such as XML and adopting Internet-based protocols. They have become a promising technology to design and build complex inter-enterprise business applications. However, Web Services are problematic to measure, control, and manage. Software metrics is vital for the management, control, and measurement of software development and despite the vast amount of techniques/mechanisms and metrics for traditional and Object-Oriented software, there has been a few research and techniques that deals with metrics for Web Services. As companies increasingly invest and relies on Web Services, the importance of metrics for those services continues to grow. In this paper I present and classify the existing metrics for Web Services, and discuss their usage and benefits. In addition, I highlight the problems found in using some of the metrics and discuss what is still lacking in this domain.

Keywords: Software Measurement, Web Service, Web Service Metrics

1. Introduction

Web Services are rapidly emerging as a popular standard for sharing data and applications over the web. Many enterprises are moving towards using and investing in Web Services in place of their traditional client-server computing and in-house servers. So Web Services are looked at as business solutions to enterprise applications integration. This observation has led to the vital requirement of having techniques and mechanisms for metrics, measurements, and models that quantify various attributes of Web Services. Metrics for traditional software applications and systems are in existence for long times before the existence and design of the Web.

One of the first and high-demanded, by service provider and their partners, class of metric for Web Services is the *quality of service (QoS)* offered by Web Services. However, because of the dynamic and unpredictable nature of the Web Services infrastructure, providing the acceptable *QoS* is really a challenging task. In addition to this, the different applications that are collaborating for Web Services interaction with different requirements will compete for network resources. The above factors will force service providers to understand and achieve Web Services *QoS*. Also, a better *QoS* for a Web Service will bring competitive advantage over others by being a unique selling point for service provider [2].

There has been a considerable amount of recent work on the challenges associated with developing metrics for web services. A large number of this work has been focused on developing, what I call, high-level or black-box based metrics, while others are focused on structural metrics. Some other work is focused on usability metrics of web sites and web services. In this paper, I will study and present the existing work that deals with high-level metrics in general and on the quality of Service metrics in particular. I will discuss the other class of metrics in another paper that is still under preparation. The rest of this paper is organized as follows: Section 2 introduces the software metrics in general. Section 3 presents the Web Services. Section 4

presents and discusses the Web Services Metrics and their classifications. Section 5 presents the main difficulties of the quality Web Services metrics. Finally, section 6 concludes the paper..

2. Software Measurement/Metrics

Software metrics is important to the management of software development, and is a mature field that has been studied for decades [3], and software measurement has become a fundamental aspect of Software Engineering. Measurement is the process by which numbers or symbols are assigned to attributes of entities in the real word in such a way as to describe them according to clearly defined rules [4]. In software measurement, the main purpose is to provide insight into software processes and products, so that any organization will better able to make decisions and manage the achievement of goals.

The terms "measure", "measurement", and "metrics" are often used interchangeably, but according to software engineering theories a ***measure*** provides a quantitative indication of the extent, amount, dimensions, capacity, or size of some attribute of a product or process. ***Measurement*** is the act of determining a measure. The IEEE Standard Glossary of Software Engineering Terms defines metrics as "a quantitative measure of the degree to which a system, component, or process possesses a given attribute".

3. Web Services

Web Services are based on a collection of standards and protocols that allow us to make processing requests to remote systems by speaking a common, non-proprietary language and using common transport protocols such as HTTP and SMTP.

Web services represent a new programming approach based on a document-oriented model designed for interoperability at a document, typically XML, level. They are modular, self-describing, self-contained applications that are based on open standards and can be published, located, and invoked across the Internet/Web. Web services enable us to build Web-based applications using any platform, object model, and programming language that we may require. In addition, they are implemented using a collection of several related, established and emerging technologies and communication protocols that include HTTP, XML, Simple Object Application Protocol (SOAP), Universal Description Discovery and Integration (UDDI), Web Services Description Language (WSDL), Common Object Request Broker Architecture (CORBA), Java Remote Method Invocation (RMI) , and .NET [1].

The web service model consists of three entities, the service provider, the service registry and the service consumer. Figure 1 shows a graphical representation of the traditional web service model:

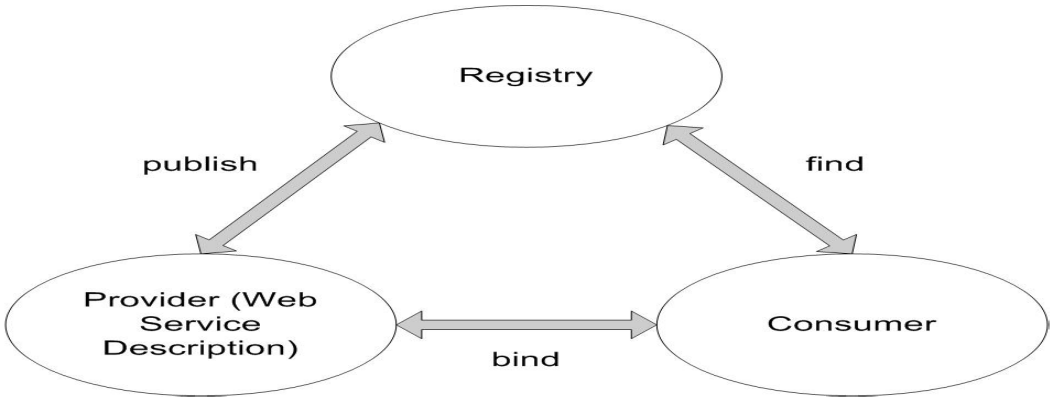

Figure 1 The web service model

The service provider creates or simply offers the web service. The service provider needs to describe the web service in a standard format, which in turn is XML and publish it in a central Service Registry. The service registry contains additional information about the service provider, such as address and contact of the providing company, and technical details about the service. The Service Consumer retrieves the information from the registry and uses the service description obtained to bind to and invoke the web service.

Web Services, like any other software applications, should have some mechanisms to measure, manage and control. This will be discussed in the next section.

4. Web Services Metrics

Although Web Services technology has been adopted by major software vendors such as IBM, Microsoft, Oracle, Borland, etc., and resulted in vast amount of research activities, there is little work on metrics for Web Services. Most of the work is focused on Quality-Of-Service. Other work is focused on projecting existing metrics for object-oriented software components [8] on Web Services systems, since web services are mostly generated from object-oriented programs. Web Service metrics are very useful for developers, providers, and users. Developers will use metrics to manage and control the best way to implement the system. Service providers can use metrics to make sure the service is running according to certain set measures, and users can use certain metrics to choose the best service providers. So metrics may differ based on who is using them, and they may have different requirement and level of importance.

Based on my finding after surveying the literatures, Web services metrics can be classified into two main classes: Structural Metrics, and Quality Metrics. In this paper, the main focus will be on the Quality Metrics.

4.1 Structural Metrics:

Under this class, most of the metrics discussed and found in the literature talk about the different types of coupling metric. Although this is not the main topic of this paper, I will summarize my finding in this section. More details about this type of metrics will be discussed in another future paper. So, most of the papers found about structural metrics talks mainly about different types of coupling metric. Coupling is the degree of interaction between services, and the basic idea of coupling metric is to count how many interactions there are in between services. In what follows, I will briefly mention the main ones related to coupling:

Choi and Lee [17] proposed a dynamic coupling metric to accurately measure the coupling between classes. Perepletchikov *et al*.[18] proposed a set of metrics for quantifying the structural coupling of design artifacts in service-oriented systems. Qian *et al*. [19] presented a practical guide for evaluating decoupling between service-oriented components in the service composition such as Business Process Execution Language (BEPL). They suggested that lower decoupled distributed software application would be much easy to understand, update and expand in the future. Quynh and Thang [20] proposed a suite of metrics to evaluate service's quality according to its ability of coupling. They showed that the coupling metrics can measure the maintainability, reliability, testability and reusability of services. Li and Henry[21] defined the Message Passing Coupling (MPC) as the count of the number of send statements that is found in methods of one class to other classes. Chidamber and Kemerer [8] introduced the Response For Class as a measure of the number of methods that can potentially be executed in response to a message received by an object of that class. In addition, they defined Coupling between Object Classes as the count of the number of classes to which it is coupled. Finally, Prasad and Nagar [22] defined a new set of operational measures for the conceptual coupling of classes.

4.2 Quality Metrics:

The Web Services Quality Metrics mainly refer to both functional as well as non-functional quality aspects of Web Services. This includes *performance*, *reliability*, *integrity*, *accessibility*, *availability*, *interoperability*, and *security*. In this section, I will present each of these aspects, then, in the second section, I will try to discuss the limitation of each of these aspects.

- **Performance:** Web Services performance measures the speed of service requests completion. It can be measured in terms of throughput, response time, latency, execution time, and some other metrics under this class. Throughput is the number of web service requests completed in a given time period. Response time is the time required to complete a web service request. Latency is the round-trip delay between sending a request and receiving the response. Enterprises cannot afford to deploy services and applications before ensuring that they perform well. Higher throughput, lower latency, lower execution and faster transaction times represent good performing Web Services. In general, high quality web services should provide higher throughput, faster response time, lower latency, and lower

execution time [2, 5]. Two types of latency are reported: connection latency and request latency. For each metric, the mean time is provided, as well as the standard deviation of all data, plus the minimum and maximum times. Connection latency reflects the time taken to establish a connection, while request latency reflects the time to complete the data transfer once the connection has been established. User-perceived latency will include the sum of connection and request latencies, plus any network latency due to WAN connections or routers. An example of performance metrics is the throughput metric called WIPS (Web Interactions Per Second) which measures the average number of Web Interactions completed per second [6].

- **Reliability:** Reliability is the ability of a web service to perform its required functions under stated conditions for a specified time interval. Reliability is the overall measure of a Web Service to maintain its service quality and it is related to the number of failures per day, week, month, or year. It also refers to the assured and ordered delivery for messages being sent and received by service requestors and service providers [7].

- **Integrity**: Integrity for web services measures to what level the services can prevent unauthorized access to, or modification of, data or programs. Program integrity measures the consistency of the results of the operation before and after the execution of the program. Data integrity measures the security of data while in transit [2].

- **Accessibility**: Accessibility measures the capability of Web Services in answering user/client requests.

- **Availability:** Availability measures the readiness of Web Services. The Web Service should be ready and available immediately when it is invoked. This availability is the probability that the system is up and related to reliability [9]. An example of this metric is the Time-to-Repair that represents the time it takes to repair the Web Service [9].

- **Scalability**: Scalability represents a measure of the capability of a web service in servicing higher number of requests or clients and operations or transactions in a given time interval [7]. It is also related to both the accessibility and the performance metrics.

- **Robustness**: Robustness measures the degree to which a web service can still perform correctly in the presence of wrong or incomplete inputs [10].

- **Interoperability:** Interoperability measures the degree to which a web service can deal and interact with different clients or other services implemented using different language and/or platforms. Web services should be interoperable between the different development environments used to implement clients and services so that developers using those services do not have to think about which programming language or operating system the services are hosted on.

- **Security:** Security metrics for Web Services measures different security aspects like non-repudiation, confidentiality, authentication, authorization, encryption, traceability, and access control. The Web Service provider may apply different levels of security policy depending on the service requestor. With the increase in the use of web services which are delivered over the public Internet, there is a growing concern about security. Each of these aspects is described in more details in [11].

- **Exception Handling:** Exception handling measures the degree to which a web service can handle an exception or an anomaly in a request or the service itself. Web services should handle all exceptions in a proper way [10]. This metric is related to the robustness and availability metrics.

5. Problems with Web Services Quality Metrics:

In this section, the shortcomings of some of the quality metrics are presented and discussed.

- **Performance:** System performance is a very complex issue where many hardware and software components are involved. When it comes to the Web Services, performance depends on several

components: application logic, network, messaging and transport protocols (e.g. SOAP and HTTP), and server hardware and software. In addition to the above, no single number can represent the performance of Web Services on all applications. This is because every Web Services' system architecture is unique in its configuration, applications, operating systems, and workload. As a result of this, Web Services exhibit a large variation in performance when running different workloads.

- **Reliability:** The Web Services currently rely on transport protocols such as HTTP, which are inherently stateless and follow a best-effort delivery mechanism. It does not guarantee whether the message will be delivered to the destination.

- **Integrity:** Transaction integrity is just one of several *Quality-of-Service* elements which are missing from the first implementation of Web Services standards of SOAP, UDDI, and WSDL.

- **Availability:** Availability of Web Services system depends on so many factors like the availability of the servers and network connections. Hence, to have a highly available Web Services system you have to have, mainly, high performance and duplicate servers, and duplicate and high bandwidth network connections. So, again this metric is a very complex one that is related to so many other components in the system.

- **Interoperability:** This metric could give different indications based on the implementation of the Web Services system, and the number of types of clients expected to be using the services. This metric should be enhanced to reflect the conformance to the set of standards used in the implementation of the Web Services so that they will have good interoperability.

- **Security:** Underlying technologies used by Web Services currently do not support role-based security features used in many Web Services systems. In addition, SOAP, the de-facto messaging standard for Web Services, does not support many security features.

6. Conclusion

Web services can be looked at as a group of closely related, emerging technologies and standards that describe a service-oriented, component-based application architecture that is based on an open, Internet-centric infrastructure and protocols. They are rapidly emerging as a popular standard for sharing data and applications over the web. Many enterprises are moving towards using and investing in Web Services in place of their traditional client-server computing and in-house servers. So they become the business solutions to enterprise applications integration. This has led to the urgent need of having metrics that quantify various attributes of Web Services. Metrics for traditional software applications and systems are in existence for long times before the existence and design of the Web; however, metrics system for Web Services is a new challenging domain that need more attention. In this paper, I have surveyed most of the existing Web Services metrics, found in the literature. My finding is that the set of Web Services metrics can be classified into two main classes: Structural metrics and Quality metrics. Most of the found metrics fall under the second class which include performance, reliability, scalability, capacity, robustness, exception handling, accuracy, integrity, accessibility, availability, interoperability, and security. This is because a better *Quality-of-Service* for a Web Service will bring competitive advantage over others by being a unique selling point for service provider [2]. So the quality metrics, and in other papers as Service Level Agreement metrics [15], are highly demanded by service provider and their partners. However, because of the dynamic and unpredictable nature of the Web Services infrastructure and the different applications that are collaborating for Web Services interaction with different requirements will compete for network and infrastructure resources, providing a set of metrics that accurately measure the quality attributes of Web Services is really a challenging task.

The Web Services introduced new and challenging software metrics problems due to their distributed and high degree of service autonomy nature and ability to use application object components from hundreds or thousands of sources. Hence, network, protocols, servers, and communication systems management mechanisms can all be involved in measuring and controlling quality attributes and metrics of Web Services systems.

7. References

[1] Mohamad Ladan, “Web Services: Technologies and Benefits” *Journal of Communication and Computer,* ISSN 1548-7709, Number 6,Volume 7, 2010.

[2] R. Sumra, D. Arulazi, "Quality of Service for Web Services-Demystification, Limitations, and Best Practices", March 2003.

[3] T. McCabe, "A Complexity Measure," *IEEE Trans. Software Eng.,* vol. 2, pp. 308-320, 1976.

[4] B. Mora & F. Garcia & F. Ruiz & M. Piattini, “Software Measurement Modeling Language”, Alarcos Research Group, Department of Computer Science, University of Castilla-La Mancha, 2008.

[5] A. Mani, A. Nagarajan, "Understanding Quality of Service for Web Services", IBM developerWorks, January 2002.

[6] Daniel A. Menasce, Virgilio A.F. Almeida , Capacity Planning for Web Services: Metrics, Models, and Methods ISBN-10: 0130659037, ISBN-13: 9780130659033 Prentice Hall, 2001.

[7] K.-C. Lee et al., “QoS for Web Services: Requirements and Possible Approaches,” World Wide Web Consortium (W3C) note, Nov. 2003.

[8] S. R. Chidamber and C. F. Kemerer, "A Metrics Suite for Object Oriented Design," *IEEE Trans. Softw. Eng.,* vol. 20, pp. 476-493, 1994.

[9] GUNTHER N. J., "The Practical Performance Analyst", McGraw-Hill, 1998.

[10] Shuping Ran, "A Model for Web Services Discovery With QoS", ACM SIGecom Exchanges, 4(1), March 2003.

[11] Mohamad Ladan, “Web Services: Security Challenges“, the proceedings of the IEEE *World Congress on Internet Security (WorldCIS-2011)*, Feb. 21-23, 2011, London, UK.

[12] M. Boxall and S. Araban, "Interface Metrics for Reusability Analysis of Components," in *ASWEC '04: Proceedings of the 2004 Australian Software Engineering Conference (ASWEC'04)*, 2004.

[13] Menasce, D. A. 2002. QoS issues in Web services. *Internet Computing, IEEE,* vol. 6, pp. 72-75.

[14] J. Norguet, “How to use Web Analytics for Improving Web Applications”, IBM, redbooks paper, 2004.

[15] Raimondi, F*., et al.* 2008. Efficient online monitoring of web-service SLAs. Proceedings of the 16th ACM SIGSOFT International Symposium on Foundations of software engineering, Atlanta, Georgia, 2008.

[16] Menasce, D. A. 2002. QoS issues in Web services. *Internet Computing, IEEE,* vol. 6, pp. 72-75

[17] Choi, M. and J. Lee. A dynamic coupling for reusable and efficient software systems. Proceedings IEEE Int’l conference on Software Engineering Research, Management and Apllications, Aug. 20-22, 2007 Busan, South Korea, pp 720-726.

[18] Perepletchikov *et al*. Coupling metrics for predicting maintainability in service-oriented designs. Proceedings of the 18th Australian Software Engineering Conference, April 10-13, 2007, Melbourne, VIC Australia, pp 329-340.

[19] Qian *et al*. Decoupling metrics for services composition. Proceeedings of the 1st IEEE/ACIS int’l conference on computer and information science, july 10-12, 2006, Honolulu, HI USA, pp 44-47.

[20] Quynh, P. T., and H. Q. Thang. Dynamic coupling metrics for service oriented software. Int’l Journal on Computer Science Engineering, 3, 2009, pp 282-287.

[21] Li, W. and S. Henry. Obect oriented metrics that predict maniatanability. Journal of System Software, 23, 1993, pp 111-122.

[22] Prasad, L. and A. Nagar. Experimental Analysis of Different Metrics of Software. Proceedings of 1st int’l conference on computational intelligence, communication systems and networks, July 23-25, 2009, Indore, pp 235-240.