Web Services Quality*

Sergio Andreozzi
Istituto Nazionale di Fisica Nucleare - CNAF and
Department of Computer Science, University of Bologna
I-40127 Bologna, Italy
sergio.andreozzi@cnaf.infn.it

Danilo Montesi
Department of Mathematics and Informatics, University of Camerino
I-62032 Camerino (MC), Italy
danilo.montesi@unicam.it

Rocco Moretti
Department of Computer Science, University of Bologna
I-40127 Bologna, Italy
rocco.moretti@cs.unibo.it

ABSTRACT

This paper deals with the organization of concepts and ideas as regards the relation between quality and web accessible services. We discuss the concept of service and service requestor. We investigate their mutual expectations and offered functionalities that involve both functional and non-functional requirements. The relation between non-functional requirements and quality attributes is presented by examining the related concepts of measurement and model. We contextualize the generic concept of service in the Web and Web services architecture and, finally, we briefly present current proposals for service functional and non-functional requirements description.

Keywords: Web Services, Quality of Service, Non-functional requirements.

1. INTRODUCTION

This paper is mainly targeted at organizing concepts and ideas as regards the relation between quality and web accessible services. Firstly, fundamental aspects concerning functional and non-functional requirements (FR's and NFR's) engineering of services are presented and, secondly, current Web technologies that are able to model these characteristics as regarded by Web accessible services are introduced. We mean 'Web accessible services' as those services that can be requested through the Web.

In order to achieve this goal, we present terms and concepts related to the involved entities (section 2); we investigate current available methodologies to identify, define, verify and audit FR and NFR (section 3); we summarize the Web accessible services evolution underlining the ongoing transition from the 'human-system' interaction towards the automatic 'system-system' interaction (section 4); finally, we review the current work-in-progress for Web services FR and NFR description (section 5).

A typical Web accessible service belonging to the human-system interaction category is an e-Business Web portal providing for access to the product/service catalogue and allowing shopping online, whereas a service belonging to the system-system interaction category in a business-to-business (B2B) context is a dynamic and automatic supplier selection based on a set of constraints on the product or service to be bought (e.g. price range, delivery time, guarantees condition, payment methods).

2. TERMS AND CONCEPTS

In this section, we present the entities that are involved in our discussion and the relationships among them. We start by introducing the basic scenario of our discussion: a 'service' being offered and a 'service requestor' that directly interact with it. The former is a coordinated set of business activities that do not generate new goods, but they

*This work is partially supported by the Istituto Nazionale di Fisica Nucleare as part of the DataTAG project, by the Italian Ministry of University (MIUR) as part of the SAHARA project and by the MURST as part of the project "Algorithms to index and query semistructured data" RSO ex-60%

consist in performing certain works, while the latter is the entity interacting with a service as a direct user.

Each of these parts offers ‘functionalities’ and has ‘expectations’ against the counterpart. By functionality we mean an action that either the service or the service requestor can do, while by expectation we mean a functionality that a service (respectively, a service requestor) expect to be provided by a service requestor (respectively, a service).

For a successful interaction between a service and a service requestor, their mutual expectations must not be greater than functionalities offered by the counterpart. Conversely, the functionalities offered by each of them must at least meet the expectations of the other one.

For instance, in the system-system scenario presented in section 1, service functionalities may consist in the catalogue searching capability that may be based on certain criteria (e.g. price range and product type) and may allow services/products purchase. In the same context, service requestor functionalities may have the following capabilities: new suppliers discovering, querying their services/products availability, and selecting the best buy with respect to some constraint. Considering expectations, service expectations may consist in the service requestor capability of presenting certain credentials or querying the catalogue using a fixed set of parameters, while service requestor expectations on the counterpart may have the service capability of enabling a query for a product on the catalogue to be satisfied within 24 hours.

Expectations and functionalities of each involved part are related to functional and/or non-functional requirements (respectively, a capability that the system must have and an aspect of a system different from its capabilities).

As anticipated above, both functionalities and expectations must be mapped in a set of FR’s and/or NFR’s concerning the service and/or service requestor involved. For instance, the service purchase functionality implies the following FR: ‘the products/services availability must be advertised’, and the following NFR: ‘the payment transaction must use a secure communication channel in order to meet goals such as confidentiality, authentication, non-repudiation and integrity’. We shall go deeper in this subject in section 3.

For the purposes of our analysis we need to introduce and define the following entities: the ‘Service Owner’ that is the entity owning the service business property, the ‘Service Provider’ that is the entity providing for service access and execution, and the ‘Service End-User’ that is the ultimate consumer of the service.

For instance, if we consider a virtual travel agency selling airline tickets of different companies, the service owner will be the company that effectively provides for the booking service, the service provider will be the travel agency acting as the intermediary, and the service end-user will be the customer buying a ticket.

We can now introduce the fundamental concept to which all the foregoing concepts given above are related, that is quality, meaning for it *the totality of features and characteristics of a product or service that bear on its ability to satisfy stated or implied needs* [1].

In section 3 we will consider quality attributes of a service/product in relationship with certain requirements and viewpoints. These viewpoints may be intended as the theoretical tool by which we can express state or implied needs.

3. QUALITY FORMALIZATION AND EVALUATION

NFR’s have been considered as ‘constraints’ [2] or ‘quality attributes’ [3]; these definitions are too restrictive for our purpose. We distinguish NFR’s from constraints because constraints mainly express features and characteristics easy to quantify. For instance, it is very difficult to define constraints and invariants concerning the ‘usability’ of a service/product. Of course, constraints may be very useful to model, for example, service/product delivery timeliness in a workflow.

The second way to consider NFR’s is as quality attributes. There are many authoritative quality attributes taxonomies [4, 5, 6] but this tendency of thought is not mainly addressed to express the aims we want to achieve by investigating a certain set of quality attributes. A quality attribute is a useful component to build taxonomies and quality models, but it does not clearly help to express how and why to employ them. For instance, to model the ‘security’ quality attribute of a service/product provider in terms of ‘confidentiality’, ‘integrity’ and ‘availability’, does not directly permit to investigate whether it is possible to guarantee the achievement of a particular access control policy.

To clearly express our quality goals and quantify the satisfaction concerning a particular quality issue, we consider NFR’s in relationship with constraints and quality attributes, but we integrate these visions with the concepts of ‘measurement’ and ‘model’.

As described in section 2, both service and service requestor have expectations on the counterpart and offer it functionalities. In their turn, both expectations and functionalities involve FR’s and/or NFR’s concerning the system under investigation.

We remark that each NFR has logic and engineering relations with certain quality attributes. Let us consider again the system-system example given in the foregoing section pointing out the described scenario. The supplier selection service offers the cited searching functionality to its requestor, while this requestor may offer the supplier discovering functionality to the service. From the expectation viewpoint, the supplier selection service may expect the requestor has certain credentials, while it expects the service guarantees a specific date of delivery.

Considering FR's and NFR's involved by the presented expectations and functionalities, for the searching functionality a FR may be that product/service availability is advertised, while for the supplier discovering requestor functionality a FR is that a supplier list is available. As regards NFR's, both the searching and supplier discovering functionalities may require a secure communication infrastructure between them. Finally, the supplier selection service expectation may imply the need for a requestor authentication technique (NFR), while the requestor expectation needs a specific delivery timeliness (NFR).

As anticipated, each NFR presented above involves directly or indirectly certain quality attributes. For instance, the specific delivery timeliness NFR relates to the 'performance' quality attribute, while the requestor authentication availability NFR concerns the 'confidentiality' quality attribute. In this context, we examine the following issues:

- How can we relate a NFR with all quality attributes concerning it according to a specific viewpoint?
- How can we quantitatively evaluate and trade off multiple quality attributes to arrive at a global system quality evaluation?

In the remaining of this section we will investigate the foregoing questions.

Viewpoints, NFR's and Quality Attributes

Quality is a complex concept that involves the entities under investigation, their quality attributes, and the observers' viewpoints. So, quality is reputed to be a not universally definable and measurable concept. Many proposals have been put forward for its definition, all of them traceable back to two tendencies of thought [7], the first one is based on the client-user of a service/product and his satisfaction, the second one on the provider/developer and his methods and tools. 'Conformity to the requirements document' is a typical definition belonging to the first tendency, while 'conformity to the design document' is a typical definition belonging to the second one.

These two tendencies of thought examine only indirectly many specific viewpoints (e.g. a catalogue service provider or a financial investor service requestor), that is there is no way to quantify explicitly the satisfaction concerning a quality characteristic of interest for a specific viewpoint involved in a service/product evaluation. Of course, we can express FR's and NFR's concerning the needs of a specific viewpoint, but in general we do not have a theoretical tool to quantify how much these requirements are respected.

Another issue to consider is that we need to define and plan for quality. In fact typical questions in this context concern, for instance, estimate of service/product cost, availability, security, and delivery timeliness. We can conclude that we must use an adaptable quality definition in terms of the involved quality attributes. Such a definition will be used by persons very different as regards their own education and goals.

Global Quality Evaluation

In this section we present the concept of 'model' and that of 'measurement'. We introduce them because the first one is an essential tool to formalize a viewpoint in terms of quality attributes it may need to consider, while the second one helps us to quantitatively evaluate and trade off multiple quality attributes and to arrive at a global quality evaluation.

Measurements:

We can interpret a measurement as an activity allowing us to describe by numbers the attributes of the service/product under examination. As regards the type of attributes they investigate, measurements are divided into two categories:

1. **Direct Measurements**. The attribute of interest is measurable without help of other measurements (e.g. the service/product delivery time).
2. **Indirect Measurements**. The attribute of interest is measurable only with the help of other measurements. For instance, the service/product delivery timeliness may be measured as the ratio between the service/product delivery time and the service/product expected delivery time; we can mean a value close to 1 as an indicator of high timeliness, while we can mean a low timeliness on the contrary.

The indirect measurements have an anticipatory power since they assume a relation among many attributes; so, because of its kind, very probably quality is measured by an indirect measurement. The need of this indirect measurement implies that we formalize the relations among the attributes of interest. In this formalization the concept of model presented in section 3 can explain the rules and the methods that allow us to define the link between a particular software/product quality viewpoint, and the entities allowing us to evaluate it.

Models

A measurement, even if defined so that it reflects the reality, cannot be used in a real analysis without a theoretical tool to interpret and employ it. Such a tool will be supplied by a model of the reality in which the measurement must be carried out. A model can be intended as an abstraction of the real world pointing out its aspects of greatest interest, that is the goals we want reach through it. We need this step to determine what aspects of the real world we must include in the model; besides, a model must lay down the relations among the involved entities and their attributes. For instance, if we want to model the paper consumption for a printing service (i.e., the goal of the model) we will assume a relation between the number of the produced lines of text and the quantity of paper it needs (i.e., the relations among the involved entities and their attributes). So, we can define the following equation (where cns is the paper consumption, $SLOC$ is the number of the produced lines of text, and n is the number of lines of text fitting in one page):

$$cns = \frac{SLOC}{n} \tag{1}$$

We remark that n is parametric on the type of the employed paper. Besides, we can also remark that a model gets its input and returns its output basing itself on relations among variables; these relations are formalized in the model by equations that may depend on parameters. The crucial point is the association between the entities of the real world and the input and output. On the one hand, we have the process of measurement that works from the real world toward the model, on the other hand we have the process of prevision and evaluation that goes in the opposite direction.

Models and Theory: Each model implies a theory expressed by the relations between its input and output; only according to this theory we can carry out previsions and evaluations. The more we require a wide applicability of a model, the more its complexity increases. Both who has chosen the simplicity to the detriment of universality [8], and who has chosen the opposite way [9, 4], have been criticized [7].

This remark induces us to consider limits of a model that are represented by the hypotheses it bases itself on. If the hypotheses are complex we have problems in the process of 'validation' of the model. This process consists in verifying the agreement of the model under examination, with the real world it claims to represent. We have essentially two possibilities to validate a model:

1. **Theoretical validation:** We build a theory the model must comply with [7]. In that case, we have three fundamental issues to consider:

   (a) The model must be usefully employed by its users that is the model must return significative information.

   (b) The model must be formal. The relations between its input and its output must be definite in all conditions that is the model must not be ambiguous.

   (c) The input of the model must be based on objective measurements. Some measurements are necessarily subjective but, in that case, we must clearly express this subjectivity.

2. **Empirical validation:** Even though a model complies with a theory, it can be averse to the reality it claims to represent. We try to verify it through statistical studies at large scale and in many environments; for instance, techniques of linear regression will help us in this context [10].

4. WEB ACCESSIBLE SERVICES

The World Wide Web was born in the early 90's as a technology for sharing scientific documents across the Internet by standard means. The architecture has been designed to meet the needs of a large-scale distributed hypermedia system. Over time, the Web which Hypertext Transfer Protocol (HTTP) [11] is the primary information transfer protocol and the Hypertext Markup Language (HTML) [12] is the primary media type, has been extended in order to provide greater support for collaboration and distributed authoring [13].

Currently, the W3 Consortium (W3C) is working on a reference document for the Web architecture [14]. In this, the Web is defined as a networked information system consisting of agents (programs acting on behalf of another person, entity, or process) that exchange information. The analysis is organized in terms of concepts such as identification (Uniform Resource Identifier or URI), representation (an Internet media type and a sequence of bits) and interaction (agents exchange representations via protocols). The concept of 'resource' is evolved from the initial *a network data object or service which can be identified by an URI* [15] to the current *anything that has identity (e.g. documents, files, menu items, machines, services, people, organizations and concepts)* [16]. This means that, in the Web, a service is part of the universe of resources, is identified by a URI and may have its own representation.

Traditional Web applications are designed to take input from a human user and display output to a human user and the Web architecture is a mature technology to accomplish this task. The new undertaken effort towards a more loosely-coupled system-system interaction, where the information exchange require that outputs of a component are intended to be in a machine-processable format, needs a specific set of technologies that must conform to the Web architecture. The Web services architecture [17] is the current proposal to meet this goal and is presented in the next section. While it is plain that we need Web services, there is not yet common view on whether they are a subset or a superset of the Web.

The Web Services Architecture

The Web services architecture is being defined inside the W3C in order to determine a common set of concepts and relationships that allow different implementations working together [17]. While no constraint on specific technologies is given, the current reference architecture assumes that implementations are build on the foundation of both Web Services Description Language (WSDL) [18] and Simple Object Access Protocol (SOAP) [19].

A Web service is defined as 'a software system identified by a URI, whose public interfaces and bindings are defined and described using XML. Its definition can be discovered by other software systems. These systems may then interact with the Web service in a manner prescribed by its definition, using XML based messages conveyed by internet protocols' [17].

The architecture is divided in a basic part that each implementation must provide and an extended part where guidelines to layer optional functionalities are given.

The basic architecture structures the minimum set of technologies that is capable of:

- Exchanging messages
- Describing Web services
- Publishing and discovering Web service descriptions

Three are the involved roles: the service provider, the service requestor and the discovery agency. They interact by exchanging messages, which may be aggregated to form message exchange pattern's (MEP's). Operations that can be undertaken are three: publish or find a Web service description and interact with a Web service.

A 'Web service description' is a machine-processable document that describes the expectations and functionalities of a particular Web service, so that a potential Client (service requestor) can read the description and understand how to correctly interact with the Service. Even though, it is written solely from the point of view of the Web service [20].

Three are the independent protocol stacks that have been defined: the 'Wire stack' that encapsulates the concepts and technologies dealing with the actual physical exchange of information between any of the roles, the 'Description stack' that allow descriptions that apply to a particular Web service or to relationships among a set of them, and the 'Discovery Agencies stack' that apply to the publish and find operations [17].

In the current reference architecture document, there are three overarching concerns that have been identified: security, management and quality of services. Security addresses issues like authorization, authentication, integrity and confidentiality. Management is defined as a set of capabilities for discovering the existence, availability, health, and usage, as well the control and configuration of resources, where resources are defined as Web services, agents of the Web services architecture, and roles undertaken in the architecture. Quality of service is not yet developed.

5. MODELLING LANGUAGES FOR QUALITY IN WEB SERVICES

Currently, the official language to describe Web services is the Web Service Description Language (WSDL) [18], a W3C specification previously developed by a number of industries and contributed to the Consortium during 2001. This proposal does not address issues related to the description of quality aspects of a service. A specific language to describe NFR's of a service has been announced by IBM and is called Web Services Endpoint Language (WSEL) [21], but, at present, any specification document has been publicly released.

Other efforts towards modelling Web services quality are the DARPA Agent Markup Language for Web Services (DAML-S) and the Web Services Offering Language (WSOL).

DAML-S is an ontology of services making possible to discover, invoke, compose, and monitor Web resources that offer particular services and have particular properties [22]. DAML-S consists of three main parts: the service profile (for advertising and discovering services), the process model (for providing a detailed description of a service operation), and the grounding (for giving details on how to interoperate with a service). In particular, the profile allows the description of properties concerning features of a service that is the category of a given service, the quality rating of the service, and a list of service parameters.

WSOL has been developed for a formal specification of various constraints and classes of service for Web services [23]. Constraints may be functional (pre-, post- and future-conditions), non-functional (Quality of Service), and access rights. Classes of service are defined as a discrete variation of the complete service and quality of service provided by a Web service.

6. CONCLUSIONS

In this paper we have investigated the relation between quality and web accessible services. Terms and concepts concerning our area of investigation (i.e. service, service requestor, expectations, functionalities, FR's and NFR's) have been presented and exemplified. We have examined the related concepts of measurement and model and we have considered the concept of service in both Web and Web services architecture concluding our discussion by presenting current proposals for service functional and non-functional requirements description.

The organization of concepts considered in this paper is the first step towards building a framework for the analysis, description, negotiation and evaluation of Web services quality.

An association between a binding and a network address, specified by a URI, that may be used to communicate with an instance of a service. An end point indicates a specific location for accessing a service using a specific protocol and data format.

References

[1] **Quality Management and Quality Assurance - Vocabulary**. ANSI/ASQ Standard.

[2] Roman G. C. **A Taxonomy of Current Issues in Requirements Engineering**. *IEEE Computer*, 18(4):14–23, 1985.

[3] Keller S. E., Kahn L. G., and Panara R. B. **Specifying Software Quality Requirements with Metrics**. In *System and Software Requirements Engineering*, pages 145–163. IEEE Computer Society Press, 1990.

[4] Boehm B., Brown J.R., Kaspar H., Lipow M., McLeod G., and Merritt M. **Characteristics of Software Quality**. TRW Series of Software Technology. North Holland, Amsterdam, 1978.

[5] **International Standard ISO/IEC 9126: Quality Characteristics and Guide Lines for their Use**. Geneva (Switzerland), 2001.

[6] McCall J. A., Richards P.K., and Walters G. F. **Factors in Software Quality**. Technical Report RADC-TR-77-363, RADC (Rome Air Development Centre), 1977.

[7] Shepperd M. and Ince D. **Derivation and Validation of Software Metrics**. In Oxford University Press, editor, *International Series of Monographs on Computer Science*, volume 9. Oxford, 1993.

[8] McCabe T. J. **A complexity measure**. *IEEE Transactions on Software Engineering*, 2(4):308–320, 1976.

[9] Boehm B. **Software Engineering Economics**. Prentice-Hall, 1981.

[10] Chulani S. **Bayesian Analysis Of Software Cost And Quality Models**, May 1999. Dissertation Presented to the Faculty of the Graduate School, University of Southern California. In partial fulfillment of the Requirements for the Degree in Doctor of Philosophy in Computer Science.

[11] Fielding R. T., Gettys J., Mogul J. C., Nielsen H. F., Masinter L., Leach P., and Berners-Lee T. **Hypertext Transfer Protocol HTTP/1.1**, 1999. Internet RFC 2616.

[12] Pemberton S. et Al. **XHTML 1.0 The Extensible HyperText Markup Language (Second Edition) - W3C Recommendation**, 2002.

[13] Fielding R. T., Whitehead E. J. Jr., Anderson K. M., Bolcer G. A., Oreizy P., and Taylor R. N. **Web-based development of complex information products**. *Communications of the ACM*, 41(8):84–92, 1998.

[14] Berners-Lee T., Bray T., Connolly D., Cotton P., Fielding R., Lilley C., Orchard D., Walsh N., and Williams S. **Architecture of the World Wide Web - W3C Working Draft**. http://www.w3.org/TR/webarch/, 2002.

[15] Berners-Lee T., Fielding R., and Nielsen H. F. **Hypertext Transfer Protocol – HTTP/1.0**, 1996. Internet RFC 1945.

[16] Berners-Lee T., Fielding R., Irvine U. C., and Masinter L. **Uniform Resource Identifiers (URI): Generic Syntax**, 1998. Internet RFC 2396.

[17] Champion M., Newcomer E., and Orchard D. **Web Services Architecture**, 2002, W3C Draft.

[18] Christensen E., Curbera F., Meredith G., and Weerawarana S. **Web Services Description Language (WSDL) 1.1**, March 2001.

[19] Box D., Ehnebuske D., Kakivaya G., Layman A., Mendelsohn N., Nielsen H. F., Thatte S., and Winer D. **Simple Object Access Protocol (SOAP) 1.1**, 2000.

[20] Liu C. K. and Booth D. **Web Services Description Language (WSDL) Version 1.2 Part 0: Primer**, September 2002.

[21] Leymann F. **Web Services Flow Language (WSFL 1.0)**, May 2001.

[22] Chinnici R., Gudgina M., Moreau J. J., and Weerawarana S. **DAML-S: Semantic Markup for Web Services**, Oct 2002.

[23] Tosic V., Pagurek B., Esfandiari B., Patel K., and Ma W. **Web Service Offering Language (WSOL) and Web Service Composition Management (WSCM)**. 2002.