Web Services Quality Assurance: A XML-Based Quality Model

José Jorge L. Dias Jr.[1,2], José Adson O. G. da Cunha[1,2], Alexandre Álvaro[1,3], Roberto S. M. de Barros[1], Silvio Romero de Lemos Meira[1,3]

[1]Centro de Informática, Universidade Federal de Pernambuco
Caixa Postal 7851 – 50.732-970 – Recife – PE – Brasil

[2]Unidade de Desenvolvimento da Paraíba, DATAPREV – Empresa de Tecnologia e Informações da Previdência Social
58.013-240 – João Pessoa – PB – Brasil

[3]C.E.S.A.R – Centro de Estudos e Sistemas Avançados do Recife
50.030-390 – Recife – PE – Brasil

{jjldj, jaogc, aa2, roberto, srlm}@cin.ufpe.br

Abstract. *The task of reusing Web services that meet quality attributes requirements is a challenging issue in service-based development approach. In such a context, a quality model can provide the means to effectively evaluation of services, enabling the certification of these services. In this sense, this paper proposes a quality model based on the ISO 9126 standard, defining a set of attributes and metrics for an effective evaluation of Web services. A XML-based representation and security schema for the model was created to support a service certification process. Moreover, a preliminary case study was elaborate in order to verify the gap between the proposed model and the available information by the Web services providers.*

1. Introduction

Component-Based Development (CBD) emerged to build systems through the composition and assembly of software components. The significant increase of abstraction and encapsulation that components provided are an important step towards service-orientation. However, their main purpose was in development reuse, while service-orientation focused on distribution and runtime reuse [Krafzig et al. 2004].

The term "service" has been present in commercial computing for a long time and has been used in many different ways. Nowadays, for example, we find large companies, such as IBM, promoting the concept of "services on demand" [Krafzig et al. 2004]. Additionally, several works point out that Service-Oriented Architectures (SOA) has influenced the development of new systems. The Gartner Institute and other authors claim that SOA proposes to be a CBD evolution [Szyperski 2002][McCoy et al. 2003].

Hence, Web Services, a SOA-based technology that has been used in industry in the last years, had emerged in order to construct applications that reuse services available in the network [Stal 2002]. A Web service is simply an application that exposes a function that is accessible using standard Web technology, such as SOAP

[SOAP 2006], XML [XML 2006] and HTTP [HTTP 2006]. This is significant because Web services are developed for and deployed onto any platform using any programming language [McGovern et al., 2003]. In this way, distributed applications development through the reused services of the Internet becomes faster, increasing the quality and decreasing product costs and delivery time [Elfatatry 2004].

Just as CBD model aims the use of artifacts developed by third parties [Szyperski 1998], the service-oriented development aims the use of services developed by third parties. In this sense, a quality model is necessary in system development through the services integration, in order to make possible the evaluation and, consequently, the reuse of services by consumers.

There are some difficulties that must be considered in the elaboration of such model, such as (i) what characteristics and quality attributes must be considered, (ii) how can we evaluate them and (iii) who must be responsible for such evaluation [Alvaro et al., 2005]. The international standards of software product quality specification, such as ISO 9126 [ISO/IEC 2001] and ISO 14598 [ISO/IEC 1998] are very general, and, thus, it is difficult to apply them in specific domains, such as Web services.

In order to apply the considered quality model, it is important to have a certification process through which the services can be evaluated. There are numerous benefits to certificate services, such as: demonstration of quality promise, consumer trust, requirements conformance and to aggregate value to the service. In this context, trust certifier entities must analyze the Web services and emit a certification for them.

This paper proposes a quality model based on the ISO 9126 standard [ISO/IEC 9126 2001], defining a set of attributes and respective metrics for an effective evaluation of Web services. Moreover, a XML-based representation was elaborated to support the proposed quality model. A security schema is also proposed in order to avoid that malicious entities corrupt the certification data, or even that, the same certification document is attached to other services. A preliminary study case was elaborate in order to evaluate if the information required for the model is actually available from some of the most widely used Web services vendors.

The remainder of this paper is organized as follows: section 2 presents related work referring to the existing quality models. Section 3 criticizes the ISO 9126 model, showing the suitable characteristics to the Web services context. Section 4 proposes the use of XML for quality model documentation, as well as a security schema to guarantee the authenticity and integrity of it. Section 5 presents a preliminary study case. Then, section 6 presents some concluding remarks.

2. Related Work

Specifications have been proposed to provide definitions and classifications of the quality characteristics of software products, like the ISO 9126 International Standard.

Because it has a high level of abstraction, it does not include in the specification a detailed set of attributes and metrics, so many studies have been realized towards a quality model for components, in the last few years [Bertoa et al. 2002][Bertoa et al. 2003][Alvaro et al. 2005], focusing the ISO 9126 model in the characteristics of such software. However, due to the particularities of Web services, it is necessary a new quality approach, that takes into account the relevant characteristics for such kind of software.

Many proposals have emerged towards Services Level Agreement (SLAs) to improve the Web services reuse by consumers [Tosic 2004]. Generally, these agreements have dynamic service information at QoS (Quality of Service) level, like performance, availability and reliability, facilitating their selection at runtime. On the other hand, as in the CBD approach, services are selected at the project life cycle. Therefore, other static quality characteristics, inherited from CBD, are important to facilitate the services selection, such as documentation, support, business model, etc.

Two works proposed to evaluate the quality of Web services. One of them is the QMWS (Quality Model for Web Services) specification, from OASIS [OASIS 2005], which provides a model to manage the Web services quality at development process and use of them. Nevertheless, there is only a conceptual level draft, being necessary the definition of the quality properties to be evaluated in accordance with the Web services characteristics. Other work was proposed by Ran [Ran 2003] and it describes a discovery model of Web services based on functional and non-functional requirements of them. In this model, the author considers a set of non-functional attributes to evaluate the quality of Web Services. But, unfortunately, it does not describe the metrics used for such evaluation, nor its real adequacy to the industry.

In this sense, this work complements such proposals, elaborating a quality model with its characteristics, sub-characteristics, attributes and metrics for Web services, evaluating not only its dynamic characteristics, but also its static characteristics.

3. Quality Model for Web Services

In accordance with [ISO/IEC 9126 2001], a quality characteristic is a set of software product properties through which its quality can be described and evaluated, being such characteristics refined in sub-characteristics.

An attribute is a property which a metric can be associated, being metric a procedure to check a product, giving a data that characterizes it.

A quality model, in turn, is a set of characteristics and sub-characteristics, as well as the relation among them, in order to provide the basis for the specification of the quality requirements, and its evaluation.

Table 1 shows characteristics and sub-characteristics that define the quality model ISO 9126. The idea proposed in this paper is to refine such model in order to accommodate particular characteristics of the Web services, defining attributes to be used by the Web services vendors, making possible the services evaluation and selection by the consumers.

Although the proposed model is based on the ISO 9126 standard, some changes have been made in order to develop a consistent model that can be used to evaluate Web services. The main changes made on ISO 9126 model are explained below.

- **Functionality:** It describes if the functions and specific properties of the Web service satisfy the consumer necessities. The Interoperability sub-characteristic was removed because was considered unnecessary to the model, since the Web services are accessed through known protocols and Internet standards. The Compatibility sub-characteristic was added in order to indicate if one given version of the Web service is compatible with its previous versions, as considered by [Bertoa et al. 2002]. The meanings of the others sub-characteristics remain the same, i.e. Accuracy describes the correct production of results; Security refers to the aptitude to prevent non authorized access; Adequacy refers to the existence of a set of appropriate functions for required tasks; and Compliance indicates if the Web services are in accordance with the conventions, norms or regulations established in relation to its functionality;

- **Reliability:** It indicates if the Web service is able to keep on performance level, throughout the time, in the established conditions. The sub-characteristics remain the same, i.e. Recoverability refers to the data reestablishment and restoration after the occurrence of faults; Fault Tolerance indicates the level of performance kept, in case of faults; and Maturity refers to the maturation status of the Web service;

- **Usability:** It refers to the effort needed for using the Web service by the consumers. The meaning of the sub-characteristics remain the same, i.e. Understandability refers to the easiness of agreement of the concepts used in the Web services; Learnability refers to the easiness of learning the Web service; and Operability refers to the easiness to operate and control the operations of the Web services;

- **Efficiency:** It describes if the involved resources and times are compatible with the performance level required by the Web service. Since the Web service is located in a place of the Internet, there is no quantitative concern of the system in relation to use resources by the Web services. For this reason, the Resource Utilization sub-characteristic is dismissible. On the other hand, the Time Behavior sub-characteristic has its name modified to Performance because it refers not only to the reply time of the service processing, but also to how fast the response is sent to the requester. The Stability sub-characteristic, in accordance with the ISO 9126, pertaining to the Manutenability sub-characteristic, by having its context modified, was relocated for the Efficiency characteristic, meaning how available is the service in a continuous and consistent form;

- **Maintainability:** It describes the easiness that software has of being modified, which include corrections, improvements or adaptations in software, due to changes in the environment, requirements or functional specifications. A natural characteristic of Web services is the fact that they are considered black-box components, that is, consumers are not worried about how the service is implemented, but what the service makes [McGovern 2003]. In this way, the developer can not make internal modifications, adaptations or reconfigurations in the Web service. For this reason, this characteristic is not included in the proposed model;

- **Portability:** It defines the easiness that software has of being transferred from an environment to another one. Since the Web service does not have the same characteristic that components to be deployable, that is, the service is available in some place of the Internet to be accessed through the network, it is not necessary that services be installed in different platforms. Therefore, the Portability characteristic is unnecessary to the model; and

- **Market:** Express the market Web services characteristics, finishing the proposed quality model. Although this characteristic is not important for the evaluation of service quality, it was incorporated to the model for making possible the analysis of some factors related to the service organization provider in order to improve service credibility for the consumers [Alvaro et al. 2005][Carvalho et al. 2006].

Table 1. Quality Characteristics of ISO 9126

Characteristics	Sub-characteristics
Functionality	Accuracy Compliance Suitability Interoperability Security
Reliability	Fault Tolerance Recoverability Maturity
Usability	Understandability Learnability Operability
Efficiency	Resource Utilization Time Behavior
Maintainability	Stability Changeability Testability
Portability	Installability Adaptability Compliance Replaceability Analyzability

Table 2 shows a summary of Web services quality model based on ISO 9126, with its modifications being divided into two classes: characteristics that are observed at runtime and characteristics that are observed during the service life cycle.

Table 2. Quality Model for Web Services

Characteristics	Sub-characteristics (Runtime)	Sub-Characteristics (Life Cycle)
Functionality	Accuracy Security	Suitability Compliance Compatibility
Reliability	Recoverability Fault Tolerance	Maturity
Usability		Understandability Learnability Operability
Efficiency	Performance Stability	

3.1. Attributes and respective metrics

Since the quality model was defined, it is necessary to specify attributes in order to make possible the characteristics evaluation. These are divided into two categories, depending if the attributes are observed at runtime or during life cycle of the Web service development.

The following types of metrics will be used to evaluate the attributes:

- **Presence:** It defines if an attribute is present or not in the Web service, consisting of a boolean value, which indicates if an attribute is present, and a string, which describes how the attribute is described by the Web service;
- **Value:** It indicates the accurate value of the Web service information, being described by an integer value and a string to indicate the unit; and
- **Ratio:** It describes percentages, being measured by an integer value in the interval of 0 to 100.

3.1.1. Attributes measured at runtime

Table 3 shows the quality attributes for Web services, which are evaluated at runtime, grouped by the sub-characteristics, and including the kind of metrics used.

Table 3. Attributes of the sub-characteristics evaluated at runtime

Sub-Characteristics (Runtime)	Attributes	Metric
Accuracy	Correctness	Ratio
Security	Data Encryption	Presence
	Controllability	Presence
	Auditability	Presence
Recoverability	Error Handling	Presence
	Persistent	Presence
Fault Tolerance	Mechanism available	Presence
Performance	Response time	Value
	Throughput	Value
	Processing Capacity	Value
Stability	Availability	Ratio
	Successability	Value
	Accessibility	Value

A description of each attribute is presented below:

- **Correctness:** It evaluates the percentage of results gotten with precision. It is calculated dividing the number of correct results by the total number of results gotten in one determined series of calls to the service;
- **Data Encryption:** It indicates if the Web service makes use of encryption in order to protect the data that it manipulates;

- **Controllability:** It indicates how the access control to the Web service is made;
- **Auditability:** It indicates if the Web service implements some auditory mechanism, registering the access of users to the service;
- **Error Handling:** It indicates if the Web service manipulates error situations.
- **Persistent:** It indicates if the Web service can store its state in a persistent way for later consultation;
- **Mechanism available:** It indicates the tolerance mechanism implemented by the Web service;
- **Response time:** It indicates the time taken to receive a reply, from a request, including the processing time and net traffic;
- **Throughput:** It indicates the amount of output that can be successfully produced during a period of time;
- **Processing Capacity:** It indicates the amount of input that can be successfully produced during a period of time;
- **Availability:** It indicates the period of time which a Web service exists or is available for use;
- **Successability:** It is defined by the number of received messages divided by the number of required messages; and
- **Accessibility:** It indicates if the Web service is accessible, through the verification if it can return an act for each required message.

3.1.2. Attributes measured at life cycle

Table 4 presents quality attributes, as well as the kind of metrics used, referring to the sub-characteristics evaluated during the Web services life cycle.

Table 4. Attributes of the sub-characteristics evaluated at life cycle

Sub-Characteristics (Life Cycle)	Attributes	Metric
Suitability	Coverage	Ratio
	Pre-condition and Post-condition	Presence
Compliance	Service agreement	Presence
	Certification	Presence
Compatibility	Compatibility	Presence
Maturity	Volatility	Value
	Evolution	Value
Understandability	Documentation available	Presence
Learnability	Time and effort to (use, configure, admin and expertise) the Web service	Value
Operability	Provided Interfaces	Value
	Dependencies	Value

- **Coverage:** It indicates how much of the required functionalities are implemented by the Web service;

- **Pre-conditions and Post-conditions:** It indicates if the Web service has pre and pos-conditions in order to accurately determine the input and output conditions of the service;
- **Service agreement:** It indicates if the Web service has a WS-* agreement, such as WSLA [WSLA 2003];
- **Certification:** It indicates if the Web service is certified by some organization.
- **Compatibility:** It indicates if the Web service is compatible with its previous versions (if it exists);
- **Volatility:** It indicates the time between the availability of a version and another one of the Web service;
- **Evolution:** It indicates the versions number of the Web service launched in the market;
- **Documentation available:** It indicates the availability of documentation, demos, user guide, APIs and tutorials;
- **Time and effort to (use, configure, admin and expertise) the Web service:** It measures the necessary time and effort for accomplishment of specific tasks, like use, configuration and administration of the Web service;
- **Provided Interfaces:** It indicates the amount of provided interfaces for the Web service; and
- **Dependencies:** It indicates the amount of services of which the service is dependent.

3.1.3. Market Attributes

The market attributes complement the model with information about the market structure that involves the service, such as:

- **Reputation:** Description of the provider to perform similar projects based on past experiences;
- **CMMi Level:** It indicates the CMMi level of the provider;
- **Support:** It describes the support mechanism offered by the provider;
- **Business model:** It indicates how the service is acquired, as well as its cost; and
- **Target Market:** It indicates the target market for which the service was implemented.

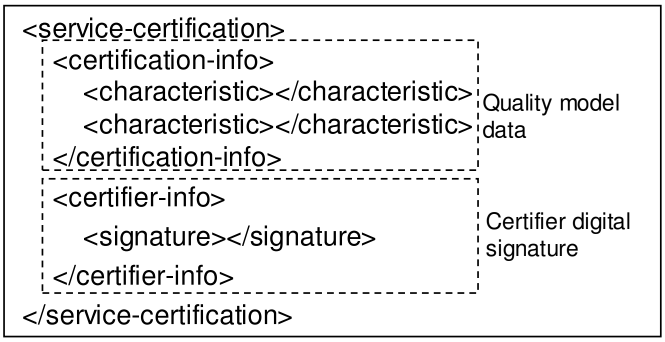
There is no specific metric to measure such attributes and, thus, they are described using a string with the information.

4. XML-Based Representation Model

Web Services architecture uses several standards used in different levels to obtain interoperability between three basic operations: publish, find and bind. Almost all these levels depend on XML language [Gottschalk 2002]. The main goal of XML use in Web Services architecture is to solve the service dynamic discovery problem [Tosic 2004].

After the quality model is defined, a certifier entity must test and validate the service, certifying it. For this reason, a XML model was developed to represent the service certification with its characteristics, sub-characteristics and attributes mentioned in section 3.

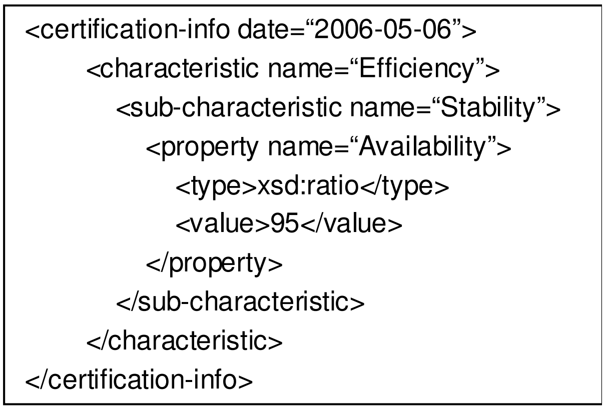
The proposed XML model is basically divided in two parts (see figure 1), where the document root is *service-certification*. The first part refers to certification data, that is, the information collected in service tests and validation. The second part mentions the certifier identify, as well as its digital signature.



```
<service-certification>
  <certification-info>
    <characteristic></characteristic>
    <characteristic></characteristic>
  </certification-info>
  <certifier-info>
    <signature></signature>
  </certifier-info>
</service-certification>
```

Figure 1. XML representation model correspond to service certification

In its first part, the model contains certification information, such as the certification date (element *date*), and elements that represent the characteristics, sub-characteristics and collected metrics attributes. Attributes are represented by the same idea proposed in [Bertoa et al. 2002], where the ODP reference model is used [ISO 10746 1997]. In this approach, each attribute is written using a *name-value* pair. Each name has a type that determines the possible values that it can assume. The benefits in this approach include the fact that it is in accordance with an international standard, and also that it is easy to document using XML templates.



```
<certification-info date="2006-05-06">
    <characteristic name="Efficiency">
      <sub-characteristic name="Stability">
        <property name="Availability">
          <type>xsd:ratio</type>
          <value>95</value>
        </property>
      </sub-characteristic>
    </characteristic>
</certification-info>
```

Figure 2. Part of the model that contains certification information

A security schema is necessary to guarantee that the service certification will not be corrupted by malicious entities. In this way, the proposed solution is based on a digital signature calculated on the XML certification model and the WSDL description URL, guaranteeing its integrity. Public key digital signatures are typically used with a hash algorithm, in order to provide integrity [Stallings 1995]. For that, the XML-Signature [W3C 2006] W3C standard was used. The main characteristic of XML-Signature is the ability to sign only specific portions of the XML tree, instead of signing the complete original document.

```
<certifier-info name=“servcertifier” url=“http://servcertifier.com/cert”>
  <Signature>
    <SignedInfo>
      <Reference URI=“http://www.services.com/quoteStock?wsdl">
        <DigestMethod Algorithm="...#sha1" />
        <DigestValue> JrtOs6lKG5fShkofd0q... </DigestValue>
      </Reference>
      <Reference URI="#certification-info">
        <DigestMethod Algorithm="...#sha1" />
        <DigestValue> LyLsF094hOpkPi4wPU... </DigestValue>
      </Reference>
    </SignedInfo>
    <SignatureValue>Hp1Z2kQLXhm5gK..</SignatureValue>
    <KeyInfo>
      <X509Data>
        <X509SubjectName>…</X509SubjectName>
        <X509Certificate>MIID5jCCA0+gA...IVN</X509Certificate>
      </X509Data>
    </KeyInfo>
  </Signature>
</certifier-info>
```

Figure 3. Part of the model that contains the digital signature of certifier entity

Figure 3 shows the second part of the certification XML document, containing the certifier information, such as *name* and *url*, and also its digital signature (element *Signature*) that follows the XML-Signature standard. The *Reference* element indicates what resources in the Web or portions of the XML must be signed. Hence, the first *Reference* element indicates the URL where the WSDL description of the service is located, and the second *Reference* mentions the XML portion that will be signed, which, in the case of the proposed model, is the content of the *certification-info*. The hash algorithm is inserted in the *DigestMethod* element and the calculated hash code from the indicated resource is in the *DigestValue* element.

Furthermore, the set of constructed references is signed, calculating the hash code of the *SignedInfo* element, signing and inserting it inside the *SignatureValue* element. In this way, it is also guaranteed integrity of the generated signature included in the references. The certificate information with certifier public key is in the *KeyInfo* element.

It is important to emphasize that the generated certificate is exclusive to the evaluated service. This is guaranteed through the signature of the reference of the WSDL description of the service. This means that any changes in the WSDL of the service will invalidate its signature.

This XML document serves as a service certification stamp, that is, the service provider will be able to attach this document to any other document that is related to the service, for example, a WSDL description or even to extend the *tModel* of the UDDI, such as considered in [Ran 2003]. In this way, the consumer locates the desired service, being able to analyze the certificate information and to verify if that certificate is signed by a trusted certifier or not. This verification is made by the consumer itself, since that

XML document contains the certifier public key. If any person tries to modify any element of the certification document, then the signature becomes invalid.

5. A Preliminary Case Study

Once we had a quality model defined, it is necessary to evaluate if the information required for the metrics in that model is actually available from some of the most widely used Web services vendors, in order to verify the current gap between the required an the provided information.

For such, Web sites that sell commercial and financial Web services were visited, namely StrikeIron [StrikeIron 2006], Xignite [Xignite 2006], and XWebServices [XWebServices 2006], that have their products published in Xmethods [Xmethods 2006], one of the most popular sites that lists publicly available Web services. From this survey, it is possible to have the first impression about the sort of information available at these sites, how it is advertised, and how difficult it is to extract the quality information we are looking for. It was not carried out a rigorous statistical analysis of the Web services population, since the objective is only to have coverage that is just enough to make a first impression of the available information possible.

Tables 5 and 6 show the information from the model that could be measured from any site analysed, confirming the gap between the required and provided information. However, this case study does not show the found attributes, but the attributes that can be measured in first sight. Therefore, the proposed quality model would make the qualitative analysis of Web services possible, allowing its use in systems development. The attributes that could not be measured in sites (i.e. any kind of information for measuring those quality attributed was not available in those site analysed) are not also shown in table 5 and 6. In tables 5 and 6, the numbers of first line represents: (1) Strikeiron, (2) Xignite, (3) XWebServices.

Table 5. Attributes (runtime) that could be measured from any site

Sub-characteristics (Runtime)	Attributes	1	2	3
Security	Controllability	Yes	Yes	Yes
Performance	Response time	No	Yes	No
Stability	Availability	Yes	Yes	No
	Accessibility	Yes	Yes	Yes
Fault Tolerance	Mechanism available	No	Yes	No

In the Xignite, the authentication is realized over SOAP Header. On the other hand, in the XWebServices, the authentication uses WS-Security conformity. Despite of the Strikeiron does not specify his authentication schema, the services just can be accessed by login process.

It is possible to verify the provided interfaces through the WSDL analysis, being also possible to verify the accessibility through tools that test the Web services. All the sites have a “Try it” option in order to verify if the service is on. Once the customers' businesses depend on the availability of the Web services, the StrikeIron and the Xignite present a graph that shows the availability of the Web services in the last days.

Table 6. Attributes (life cycle) that could be measured from any site

Sub-characteristics (Life Cycle)	Attributes	1	2	3
Compliance	Service agreement	No	No	Yes
Compatibility	Compatibility	No	No	Yes
Maturity	Evolution	Yes	No	Yes
Understandability	Documentation available	Yes	Yes	Yes
Operability	Provided interfaces	Yes	Yes	Yes

In order to provide such availability, the Xignite implements redundancy in all the levels of its infrastructure, thus guaranteeing the fault tolerance. However, there is not information about what mechanism is used to guarantee this feature.

In the XWebServices is possible verifies the service evolution, since it keeps all versions of service. Besides, it indicates the changes occurred in previous versions. The Strikeiron have only the version of the Web service, to allow getting a notion of its evolution.

About documentation available, all the sites have a several kind of information about the services, such as: operations description, client examples, message examples, demos, etc. By the other hand, in relation to the market attributes, it is possible to get, in all the sites, information about the Web service provider, support types, business models and the target market.

In general, several attributes can be measured with specific tests by service certifiers, allowing a complete Web service certification.

6. Concluding Remarks

A quality model based on the ISO 9126 international standard was presented in this work in order to help the service certification. The goal of the proposed model is not limited to services dynamic selection, as it also aims to facilitate, in a simple way, a static choice of the most appropriate service to be reused by the consumer.

In order to support the proposed quality model, a XML-based representation model was developed. Besides, a security schema using the XML-Signature standard was elaborate in order to guarantee the integrity and authenticity of the certificate generated by the certifier entity.

The lack of Web service information by the providers was verified in the preliminary study case. In this way, this model could be adopted by Web services vendors that publish Web services, guaranteeing information standardization in these sites, making possible the services selection by consumers in the system development adopting the SOA approach.

As future work, a formal case study about the services information available in the vendor sites will be done, in order to verify how much information of the model is actually available in the Web services providers. Furthermore, it will be developed a technology infrastructure in order to support the Web services certification with the proposed XML-based quality model.

References

Alvaro, A., et al, (2005). Quality Attributes for a Component Quality Model, In The 10Th International Workshop on Component-Oriented Programming (WCOP), in Conjunction with the 19Th European Conference on Object Oriented Programming (ECOOP), Glasgow, Scotland.

Bertoa, M., Troya, J., Vallecillo, A., (2003). A Survey on the Quality Information Provided by Software Component Vendors. In Proceedings of QAOOSE 2003 at 7th ECOOP.

Bertoa, M., Vallecillo, A, (2002). Quality Attributes for COTS Components. Proceedings of the 6th International ECOOP Workshop on Quantitative Approaches in Object-Oriented Software Engineering (QAOOSE), Spain.

Carvalho, J. P., Franch, X., (2006). Extending the ISO/IEC 9126-1 Quality Model with Non-Technical Factors for COTS Components Selection. ICSE. China.

Elfatatry, A., Layzell, P., (2004). Negotiating in Service-Oriented Environments. Communications of the ACM. Vol 47, Nº 8.

Gottschalk, K., Graham, S., Kreger, H., Snell, J., (2002). Introduction to Web Services Architecture. IBM System Journal 41, nº 2.

HTTP, (2006). Hypertext Transfer Protocol. http://www.w3.org/Protocols/rfc2616/rfc2616.html.

IBM Corporation, (2003). Web Service Level Agreement (WSLA) Language Specification. Available at http://www.research.ibm.com/wsla/WSLASpecV1-20030128.pdf.

ISO 14598, (1998). Software Engineering – Product evaluation. Internacional Standard ISO/IEC 14598, Internacional Standard Organization.

ISO 9126, (2001). Information Technology – Product Quality – Part1: Quality Model, International Standard ISO/IEC 9126, International Standard Organization.

ISO/IEC. RM-ODP, (1997). Reference Model for Open Distributed Processing. Rec. ISO/IEC 10746-1 to 10746-4, ITU-T X.901 to X.904, ISO/ITU-T.

Krafzig, D., Banke, K., Slama, D., (2004). Enterprise SOA: Service-Oriented Architecture Best Practices. Prentice Hall PTR.

McCoy, D. W., Natis, Y. V., (2006). Service-Oriented Architecture: Mainstream Straight Ahead. http://www4.gartner.com/DisplayDocument?doc_cd=114361

McGovern, J., Tyagi, S., Stevens, M., Matthew, S., (2003). Java Web Services Architecture. Morgan Kaufmann Publishers.

OASIS, (2006). Web Services Quality Model (WSQM). Available at http://www.oasis-open.org/committees/download.php/15910/WSQM-ver-2.0.doc.

Ran, S., (2003). A Model for Web Services Discovery With QoS. ACM SIGecom Exchanges, Vol. 4.

Simple Object Access Protocol (SOAP), (2006). http://www.w3.org/TR/soap/.

Stal, M., (2002). Web Services: Beyond Component-Based Computing. Communications of the ACM, Vol. 45, No 10.

Stallings, W., (1995). Network and Internetwork Security: Principles and Practice. Prentice Hall.

StrikeIron. (n.d.). Retrieved November 4, (2006), from http://www.strikeiron.com.

Szyperski, C., (2002). Component Software: Beyond Object-Oriented Programming. Addison-Wesley.

Tosic, V., Pagurek, B., (2004). On Comprehensive Contractual Descriptions of Web Services. Technical Report, Department of Computer Science, The University of Western Ontario.

W3C World Wide Web Consortium, (2006). XML Signature WG. http://www.w3.org/Signature/.

Xignite. (n.d.). Retrieved November 4, (2006), from http://www.xignite.com.

XMethods. (n.d.) Retrieved November 03, (2006), from http://www.xmethods.com.

XML - Extensible Markup Language. (n.d.). Retrieved November 11, (2006), from http://www.w3.org/XML/.

XWebServices. (n.d.). Retrieved November 4, (2006), from http://www.xwebservices.com.