

Centro Nacional de Investigación y Desarrollo Tecnológico

Subdirección Académica

Departamento de Ciencias Computacionales

TESIS DE MAESTRÍA EN CIENCIAS

Estudio de Atributos de Calidad de Servicios Web

presentada por

Ing. Carlos López Vázquez

como requisito para la obtención del grado de
Maestro en Ciencias en Ciencias de la Computación

Director de tesis

Dra. Olivia Graciela Fragoso Díaz

Codirector de tesis

Dr. René Santaolaya Salgado

Cuernavaca, Morelos, México. 14 de marzo de 2014.

SEP

SECRETARÍA DE
EDUCACIÓN PÚBLICA



Subsecretaría de Educación Superior
Dirección General de Educación Superior Tecnológica
Centro Nacional de Investigación y Desarrollo Tecnológico

"2013, Año de la Lealtad Institucional y Centenario del Ejército Mexicano"

Cuernavaca, Mor., 21/noviembre/2013

OFICIO No. DCC/090/2013

Asunto: Aceptación de documento de tesis

C. DR. GERARDO V. GUERRERO RAMÍREZ
SUBDIRECTOR ACADÉMICO
PRESENTE

Por este conducto, los integrantes de Comité Tutorial del **C. Carlos López Vázquez**, con número de control M10CE043, de la Maestría en Ciencias en Ciencias de la Computación, le informamos que hemos revisado el trabajo de tesis profesional titulado "**Estudio de Atributos de Calidad de Servicios WEB**" y hemos encontrado que se han realizado todas las correcciones y observaciones que se le indicaron, por lo que hemos acordado aceptar el documento de tesis y le solicitamos la autorización de impresión definitiva.

DIRECTORA DE TESIS

Dra. Olivia Graciela Fragozo Díaz
Doctora en Ciencias en Ciencias de la Computación
7420199

CO-DIRECTOR DE TESIS

Dr. René Santaolaya Salgado
Doctor en Ciencias de la Computación
4454821

REVISOR 1

Dr. Juan Carlos Rojas Pérez
Doctor en Ciencias en Ciencias de la Computación
6099372

REVISOR 2

M.C. Mario Guillén Rodríguez
Maestro en Ciencias con Especialidad en Sistemas
Computacionales
7573768

REVISOR 3

M.C. Humberto Hernández García
Maestro en Ciencias con Especialidad en Sistemas
Computacionales
7573641

C.p. Lic. Guadalupe Garrido Rivera - Jefa del Departamento de Servicios Escolares.
Estudiante
Expediente

AMR/lmz

cenidet[®]
Centro Nacional de Investigación
y Desarrollo Tecnológico

Interior Internado Palmira S/N, Col. Palmira, C.P. 62490 Cuernavaca, Mor.
Tels. (01)777 362-77-71, Conmut. (01)777 362-77-70 Ext. 414, e-mail: direccion@cenidet.edu.mx
www.cenidet.edu.mx



Cuernavaca, Mor., 06 de marzo de 2014
OFICIO No. SAC/159/2014

Asunto: Autorización de impresión de tesis

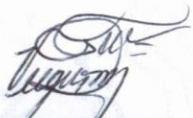
ING. CARLOS LÓPEZ VÁZQUEZ
CANDIDATO AL GRADO DE MAESTRO EN CIENCIAS
EN CIENCIAS DE LA COMPUTACIÓN
PRESENTE

Por este conducto, tengo el agrado de comunicarle que el Comité Tutorial asignado a su trabajo de tesis titulado **"Estudio de atributos de calidad de servicios WEB"**, ha informado a esta Subdirección Académica, que están de acuerdo con el trabajo presentado. Por lo anterior, se le autoriza a que proceda con la impresión definitiva de su trabajo de tesis.

Esperando que el logro del mismo sea acorde con sus aspiraciones profesionales, reciba un cordial saludo.

ATENTAMENTE

"CONOCIMIENTO Y TECNOLOGÍA AL SERVICIO DE MÉXICO"



DR. GERARDO VICENTE GUERRERO RAMÍREZ
SUBDIRECTOR ACADÉMICO



CENTRO NACIONAL DE
INVESTIGACIÓN Y
DESARROLLO
TECNOLÓGICO
SUBDIRECCIÓN
ACADÉMICA

C.p. Lic. Guadalupe Garrido Rivera.- Jefa del Departamento de Servicios Escolares.
Expediente

GVGR/mcr

Dedicatorias

A Dios:

A dios por darme la oportunidad de cumplir mis metas y objetivos.

A mis padres:

Por ser mi inspiración y guías en este recorrer de mi vida. Sin ustedes esto no habría sido posible.

A mis hermanos:

Por su apoyo y por ser los mejores hermanos del mundo, porque cada triunfo es pensando en ustedes y para ustedes.

Agradecimientos

A Dios por permitirme cumplir mis sueños y completar mis metas.

A mis padres y hermanos, por el apoyo incondicional que me han brindado, porque son la inspiración de cada uno de mis días, por su comprensión y ánimo para llegar a la meta.

A CONACYT por el apoyo económico para el desarrollo de esta tesis de maestría.

Al Centro Nacional de Investigación y Desarrollo Tecnológico CENIDET por permitirme pertenecer a esta institución y realizar mis estudios de maestría.

A mi director y codirector de tesis: la Dra. Olivia G. Fragoso Diaz y el Dr. René Santaolaya Salgado, por el apoyo brindado durante el desarrollo de esta tesis y sobre todo por brindarme su amistad. Muchas gracias.

Al comité revisor: M. C. Humberto Hernández García, Dr. Juan Carlos Rojas Pérez y al M.C. Mario Guillen Rodríguez, por el tiempo brindado a la revisión de esta tesis y por sus sugerencias y valiosas aportaciones.

Resumen

En el presente trabajo de investigación se establecen las bases que permiten obtener un marco de métricas para la medición de los atributos de calidad de Servicios Web con el propósito de obtener un marco de medidas de posibles valores estándares de calidad que ayuden a la toma de decisiones en la selección, construcción de algún Servicio Web o cuando este forme parte de una composición.

Como resultado del estudio de los atributos de calidad de WS se presenta una taxonomía. Esta taxonomía describe el análisis de los trabajos encontrados, lo que permite realizar una primera clasificación de atributos de calidad, ubicándolos en las áreas de selección, composición y reutilización. Así mismo, se describen enfoques recientes como son: de medición del nivel del Servicio, relacionados con el negocio y seguridad.

No todos los atributos de calidad cuentan con una métrica, en su lugar, sólo se cuenta con la definición o concepto de tal atributo, aunque, se observa que los atributos predominantes en selección y composición de servicios son: “*disponibilidad*”, “*seguridad*”, “*rendimiento*”, “*confiabilidad*”, “*precio/costo*” y “*tiempo de respuesta*”. Si bien la cantidad de atributos de calidad identificados no es exhaustiva, en este trabajo de investigación se presenta una cantidad considerable de atributos de calidad junto con sus fórmulas.

Abstract

In this thesis are set out the basis which allow to obtain a metrics frame for measuring Web Services quality attributes in order to obtain a measure frame of possible quality standard values which help in decision making in the selection, in setting up some Web Service or when this service becomes part of a composition.

A taxonomy as a result of the study of the quality Web Services attributes is presented. This taxonomy shows the analysis of papers found and allows to carry out a first classification of quality attributes, placing them on selection, composition and reuse areas. Likewise, recent approaches are described as follows: service level measurement, related with business and security.

Not all quality attributes have a metric, instead, we only have its definition or concept. Nevertheless, it is observed that the predominant attributes in selection and composition are: availability, security, throughput, reliability, price/cost and response time. Even if the amount of quality attributes identified is not exhaustive, in this thesis is presented a considerable amount of quality attributes together with their formulas.

Tabla de Contenido

Página

TABLA DE CONTENIDO	I
LISTADO DE FIGURAS	IV
LISTADO DE TABLAS	V
GLOSARIO	VI
CAPÍTULO 1: ANTECEDENTES.....	1
1.1 Introducción.....	2
1.2 Antecedentes	2
1.2.1 Banco de pruebas orientado a la calidad o QoS para apoyar la selección y composición de Servicios Web [Bejar, 2009]	2
1.2.2 Evaluación de medidas de similitud aplicadas a la selección de Servicios Web [González, 2011]	3
1.2.3 Marco orientado a objetos para cálculos de similitud [Valenzuela, 2012]	3
1.3 Planteamiento del problema.....	4
1.4 Objetivo	4
1.5 Producto resultado y beneficios.....	4
1.6 Alcances y limitaciones	4
1.7 Organización del documento	5
CAPÍTULO 2: MARCO CONCEPTUAL	6
2.1 Servicios Web.....	7
2.2 Protocolos Estándar de los Servicios Web	7
2.2.1 SOAP	8
2.2.2 WSDL	8
2.2.3 HTTP	8
2.2.4 XML	8
2.2.4.1 Propósito de XML.....	9
2.2.5 UDDI	9

2.3 QoS (Calidad de Servicio)	10
2.4 Medición, Medida y Métrica	10
2.5 Definiciones de Cloud y Grid Computing	10
2.6 Algunas definiciones de “Grid Computing”	11
 CAPÍTULO 3: TRABAJOS RELACIONADOS	 12
3.1 Trabajos relacionados	13
3.1.1 Propuestas para incluir atributos de calidad en los WS	13
3.1.2 Selección de Servicios Web	14
3.1.3 Composición de Servicios Web	15
3.1.4 Composición de Servicios Web y procesos de negocio	16
3.1.5 Reutilización de Servicios Web	16
 CAPÍTULO 4: PRESENTACIÓN DEL ESTADO DEL ARTE.....	 20
4.1 Taxonomía	22
4.2 Selección de WS	23
4.2.1 Selección de WS mediante Agentes y/o algoritmos de selección	23
4.2.2 Descubrimiento de WS mediante mecanismos de selección	25
4.2.3 Monitoreo de propiedades de QoS	26
4.2.4 Selección de Servicios Web en otros dominios	27
4.2.5 SLA	28
4.3 Composición de WS	29
4.3.1 Composición de WS mediante flujos de composición	29
4.3.2 Composición de WS usando Algoritmos	30
4.3.3 Composición de WS mediante agentes o middleware	31
4.3.4 Composición de servicios bajo otros dominios	31
4.4 Reutilización de WS.....	32
4.5 QoS de WS en la nube computacional.....	32
4.5.1 Acerca de GRID y Cloud Computing	32
4.5.2 Acerca de seguridad en Cloud Computing	36
4.5.3 Selección de Servicios Web en Cloud Computing	39
4.5.4 Composición de WS en Cloud Computing	40
4.6 Otra información relevante a trabajos sobre QoS de WS	41
4.6.1 Actividad del grupo de trabajo de la W3C	41
4.6.1.1 ¿Qué es la W3c?	41
4.6.1.2 Acerca de los grupos de trabajo de la W3C	41

4.6.1.3 Estado de actividad de los Servicios Web	42
4.6.2 Modelo de calidad de WS	42
4.6.2.1 WSQM	42
4.6.2.2 Calidad de Servicios Web como un Servicio	43
4.7 Atributos de calidad y Métricas.....	46
4.7.1 Conceptos o definiciones de atributos de calidad	46
Según [OASIS, 2005] y [OASIS, 2010] se definen los siguientes atributos de calidad:	50
Según [O'Brien, 2005] se definen los siguientes atributos de calidad:	52
4.7.2 Atributos de calidad & Métricas en selección de Servicios Web	53
4.7.3 Atributos de calidad & Métricas en Composición de Servicios Web	64
CAPÍTULO 5: CONCLUSIÓN Y TRABAJOS FUTUROS.....	73
5.1 Conclusiones	74
5.2 Trabajos futuros.....	76
ANEXO A: MODELOS DE CALIDAD DE SOFTWARE	77
A.1 MODELOS DE CALIDAD DE SOFTWARE	78
REFERENCIAS.....	82

Listado de Figuras

FIG. 1: SERVICIOS WEB EN FUNCIONAMIENTO [W3C, 2010].	7
FIG. 2: TAXONOMÍA DE ATRIBUTOS DE CALIDAD DE WS.	22
FIG. 3: MODELO DE CALIDAD DE SERVICIOS WEB, WSQM DE OASIS [OASIS, 2005].	43
FIG. 4: FACTORES DE CALIDAD DE SERVICIOS WEB [OASIS, 2005].	44
FIG. 5: ESTRUCTURA DE FACTOR DE CALIDAD DE WS.	46
FIG. 6: SUB-FACTORES DE CALIDAD.	51
FIG. 7: ATRIBUTOS DE CALIDAD RELACIONADOS CON "PERFORMANCE" [OBERORTNER 2011].	59
FIG. 8: PATRONES PARA MEDICIÓN DE ATRIBUTOS DE CALIDAD RELACIONADOS CON "PERFORMANCE" [OBERORTNER 2011, OBERORTNER 2010, RAJENDRAN 2009, ROSENBERG 2006].	59
FIG. 9: CLASIFICACIÓN DE ATRIBUTOS DE CALIDAD PARA WS.	60
FIG. 10: MODELO DE CALIDAD PARA CALIDAD INTERNA Y EXTERNA [MORENO, 2007].	80
FIG. 11: MODELO DE CALIDAD PARA CALIDAD DE USO [MORENO, 2007].	80

Listado de Tablas

TABLA 1: TABLA COMPARATIVA DE TRABAJOS RELACIONADOS.....	18
TABLA 2: TABLA COMPARATIVA DE TRABAJOS RELACIONADOS (CONTINUACIÓN).	19
TABLA 3: GRID COMPUTING VS CLOUD COMPUTING [HASHEMI, 2012].	35
TABLA 4: ATRIBUTOS DE CALIDAD Y MÉTRICAS [LEE 2010].	53
TABLA 5. ATRIBUTOS DE CALIDAD Y MÉTRICAS [CARDOSO, 2004], [ZENG, 2003]. [HWANG, 2007]	53
TABLA 6. ATRIBUTOS DE CALIDAD Y MÉTRICAS [YU, 2007].....	54
TABLA 7. ATRIBUTOS DE CALIDAD Y MÉTRICAS [YU, 2005].....	55
TABLA 8. ATRIBUTOS DE CALIDAD Y MÉTRICAS [LIU, 2004].....	55
TABLA 9. ATRIBUTOS DE CALIDAD Y MÉTRICAS [PATEL, 2003].	55
TABLA 10. ATRIBUTO DE CALIDAD Y MÉTRICAS [KALEPU, 2003].	56
TABLA 11: ATRIBUTOS DE CALIDAD Y MÉTRICAS [ZENG, 2003],[ZENG, 2004].	57
TABLA 12: ATRIBUTOS DE CALIDAD Y MÉTRICAS [RAJENDRAN 2009].	57
TABLA 13: ATRIBUTOS DE CALIDAD Y MÉTRICAS [ZHENG 2010].....	57
TABLA 14: ATRIBUTOS DE CALIDAD Y MÉTRICAS RELACIONADAS CON "PERFORMANCE" [OBERORTNER 2011], [OBERORTNER 2010], [RAJENDRAN 2009] Y [ROSENBERG 2006].	58
TABLA 15: ATRIBUTOS DE CALIDAD Y MÉTRICAS RELACIONADAS CON "DEPENDABILITY" [ROSENBERG 2006].	58
TABLA 16: ATRIBUTOS DE CALIDAD Y MÉTRICAS [ZENG, 2003]	64
TABLA 17: FUNCIONES DE AGREGACIÓN PARA CALCULAR QOS DE WS COMPUESTOS [ZENG 2003].	64
TABLA 18: ATRIBUTOS DE CALIDAD Y MÉTRICAS [CHEN 2006].....	64
TABLA 19: ATRIBUTOS DE CALIDAD Y FUNCIONES DE AGREGACIÓN [CANFORA 2005].....	65
TABLA 20: ATRIBUTOS DE CALIDAD Y FUNCIONES DE AGREGACIÓN [YOU 2005].	65
TABLA 21: ATRIBUTOS DE CALIDAD Y FUNCIONES DE AGREGACIÓN [ZENG 2004].....	65
TABLA 22: ATRIBUTOS DE CALIDAD Y MÉTRICA [MENASCÉ 2002].....	66
TABLA 23: ATRIBUTOS DE CALIDAD Y MÉTRICA [MENASCÉ 2004].....	66
TABLA 24: ATRIBUTOS DE CALIDAD Y MÉTRICAS [LI 2009].	66
TABLA 25: ATRIBUTOS DE CALIDAD Y FUNCIONES DE AGREGACIÓN [CARDOSO 2004].	66
TABLA 26: ATRIBUTOS DE CALIDAD Y FUNCIONES DE AGREGACIÓN [CARDOSO 2004].	67
TABLA 27: ATRIBUTOS DE CALIDAD Y FUNCIONES DE AGREGACIÓN [CARDOSO 2004].	67
TABLA 28: ATRIBUTOS DE CALIDAD Y FUNCIONES DE AGREGACIÓN.....	71
TABLA 29: ATRIBUTOS DE CALIDAD Y FUNCIONES DE AGREGACIÓN.....	71

Glosario

WS	Web Service(s)
QoS	Quality of Services, Calidad de servicio.
UDDI	Universal Description, Discovery and Integration.
WSDL	Web Services Description Language.
SOAP	Simple Object Access Protocol.
XML	eXtensible Markup Language.
HTTP	Hypertext Transfer Protocol Secure.
VM	Virtual Machine o Máquina virtual.
Hypervisor	Programa que permite que multiples sistemas operativos compartan un único host.
IT / TI	Information technology o tecnologías de la informacion.
DB	Data Base / Base de Datos.
SO	Sistema Operativo.
ACID	Atomicity, Consistency, Isolation, Durability.
NIST	Instituto Nacional de Estándares y Tecnología de los EUA.

CAPÍTULO 1: Antecedentes

En este capítulo se describen y definen las bases para el desarrollo del presente trabajo de investigación. El capítulo atiende los temas: Introducción, Antecedentes, Contexto de la tesis, Planteamiento del problema, Objetivo, Producto resultado y beneficios, Alcances y limitaciones.

1.1 Introducción

Los Servicios Web son las nuevas formas de software de Internet que pueden ser universalmente implementados e invocados con tecnologías estándar tal como: XML, SOAP, WSDL, HTTP entre otras. La calidad del servicio o QoS (de las siglas en inglés Quality of Service) está dada por un conjunto de atributos de calidad. La necesidad de las especificaciones de atributos de calidad en los Servicios Web es impulsada por dos demandas. El consumidor del Servicio Web por experimentar el buen desempeño del servicio y por la otra parte, cuando esto llega a los negocios electrónicos, los proveedores de servicios necesitan formular ofertas compatibles con QoS a fin de obtener el mayor beneficio posible de su negocio.

Los Atributos de Calidad de Servicios, se utilizan ampliamente en propuestas de solución a problemas que se presentan en las áreas de selección, composición y reutilización de Servicios Web.

En el trabajo de investigación se realiza un estudio de atributos de calidad de los Servicios Web, y de las diferentes formas de medirlos, para ello, se identificaron las medidas o métricas asociadas a los atributos de calidad de los Servicios Web, que estén relacionadas directamente con el servicio más que con la infraestructura que lo soporta.

1.2 Antecedentes

En el área de Ingeniería de Software del Centro Nacional de Investigación y Desarrollo Tecnológico (CENIDET) se ha iniciado con la investigación de los atributos de calidad de servicios con el desarrollo de tres tesis, la primera implementó un banco de pruebas orientado a la calidad o QoS de Servicios Web, la segunda se enfoca en la evaluación de medidas de similitud aplicadas a la selección de Servicios Web y la tercera es un marco orientado a objetos de medidas de similitud. A continuación se presenta una breve descripción:

1.2.1 Banco de pruebas orientado a la calidad o QoS para apoyar la selección y composición de Servicios Web [Bejar, 2009]

En este trabajo de tesis se genera un banco para pruebas de selección de Servicios Web en dos dominios, el de estadística descriptiva y el de predicción climatológica. El banco de pruebas proporciona descripciones de calidad en el WSDL de los Servicios Web para que modelos de selección por atributos de calidad sean probados y evaluados.

También se describe el sistema WeSSQoS como medio para probar el banco de pruebas, los WS se priorizan según su grado de satisfacción de los requisitos no funcionales, calculable a partir de un conjunto de atributos de calidad de dichos Servicios Web, que pueden declararse en

el WSDL o bien calcularse dinámicamente mediante monitorización. El término “monitorización” es utilizado en éste trabajo, ya que mediante la monitorización se realiza la revisión de los valores de los atributos de calidad de los Servicios Web.

La información acerca de los atributos de calidad puede provenir de diversas fuentes “diferentes repositorios WSDL, diferentes monitores, Bancos, etc.”, para probar el funcionamiento de WeSSQoS se utiliza el banco de pruebas generado y algunos monitores. La arquitectura de WeSSQoS permite la coexistencia de diversos algoritmos de ordenamiento de los WS, si bien esta tesis se centra en uno de ellos que usa la distancia euclidiana como criterio de ordenación.

1.2.2 Evaluación de medidas de similitud aplicadas a la selección de Servicios Web [González, 2011]

En este trabajo de tesis se analiza el comportamiento de siete medidas de similitud de los enfoques: espacio vectorial, correspondencia de características y heurísticos.

La similitud es un factor importante que sirve como principio de organización y clasificación, es por ello que en este trabajo se evaluó el comportamiento de las medidas de similitud, con el propósito de identificar cuáles son las que reflejan mejor la similitud entre las características de un servicio y la solicitud de un usuario expresada en atributos de calidad de servicio.

1.2.3 Marco orientado a objetos para cálculos de similitud [Valenzuela, 2012]

En este trabajo de tesis el producto principal es un marco orientado a objetos de medidas de similitud, implementando abstracciones y considerando los patrones de diseño Template Method y Strategy como su diseño estructural, haciéndolo extensible y adaptable, cuyo objetivo es apoyar a los usuarios a acelerar el proceso de desarrollo, reutilizar el código implementado en el marco y aumentar la calidad del código.

Algunas de las medidas de similitud que el marco implementa son aquellas basadas en: coeficientes de correlación, considerando datos binarios y continuos, los indicadores de distancia que consideran a los individuos como vectores en el espacio geométrico, así como coeficientes para cálculos de similitud semántica considerando las ontologías como recurso principal.

Las pruebas se realizaron para demostrar el funcionamiento del marco bajo el enfoque de resultados esperados contra resultados obtenidos.

1.3 Planteamiento del problema

Actualmente en los procesos de construcción, selección, así como en la composición de Servicios Web se propone el uso o empleo de atributos de calidad como base para estos procesos. Esto es importante tanto para las empresas que ofrecen Servicios Web, como también para los que consumen los WS¹. Sin embargo, no se tiene identificado como medir esos atributos de calidad. En consecuencia, los valores de los atributos de calidad son arbitrarios en tales procesos, representando imprecisión en los requerimientos y por consecuencia en la búsqueda de los elementos que puedan representar una solución para esos requerimientos. Adicionalmente, no en todas las ocasiones se propone el atributo de calidad junto con su forma de medirlo.

1.4 Objetivo

El objetivo de este proyecto es establecer las bases que permitan obtener un marco de métricas para la medición de los atributos de calidad de Servicios Web para después obtener un marco de medidas de posibles valores estándares de calidad que ayuden a la toma de decisión en la selección, construcción de algún Servicio Web o cuando este forme parte de una composición.

1.5 Producto resultado y beneficios

El resultado de este trabajo de tesis es un documento del estado del arte en el cual se analiza un conjunto de atributos de calidad y un conjunto de fórmulas para medirlos, con las que se puedan obtener datos que puedan ser utilizados como marco de valores de calidad, para cuando se construya, seleccione o se requiera que un Servicio Web participe en un proceso de composición, así como también, para que los usuarios y proveedores de Servicios Web puedan entender a los atributos de calidad y sus formas de medirlos.

1.6 Alcances y limitaciones

Los alcances establecidos para este proyecto de tesis son:

- El estudio de los atributos de calidad de Servicios Web.
- La identificación de medidas o métricas asociadas a los atributos de calidad, estas tienen que ser relacionadas directamente con el servicio más que con la infraestructura que lo soporta.

Las limitaciones del proyecto de tesis son:

¹ De aquí en adelante se usará el término “WS” para referirse a Servicios Web.

- La cantidad de atributos de calidad revisados no es exhaustiva.
- Se utilizan medidas existentes en la literatura.
- No se propone un nuevo atributo de calidad.
- No se propone una nueva medida o métrica para el atributo de calidad.

1.7 Organización del documento

El resto del documento está organizado como sigue:

Capítulo 2 Marco Conceptual. En este capítulo se presentan definiciones de los conceptos que son utilizados como soporte de éste trabajo de investigación.

Capítulo 3 Trabajos Relacionados. En este capítulo se describen los trabajos relacionados encontrados en la literatura actual que tratan la importancia de la gestión de los atributos de calidad.

Capítulo 4 Presentación del estado del arte. En este capítulo se discutirán los trabajos encontrados en la literatura actual que abordan la medición de atributos de calidad de Servicios Web dentro de los rubros de selección, composición y reutilización. Por otra parte se aborda la actividad del grupo de trabajo de la W3C, un modelo de calidad de WS, así como los atributos de calidad en la nube.

Capítulo 5 Conclusión y trabajos futuros. En este capítulo se describen las conclusiones de la tesis. Así como también se presentan los trabajos futuros resultantes de esta investigación.

CAPÍTULO 2: Marco conceptual

En este capítulo se describen los Servicios Web, su funcionamiento y sus estándares. También se presentan definiciones de los conceptos que son utilizados para el desarrollo de esta tesis. Se definen conceptos tales como: WS, protocolos estándar de los WS, SOAP, HTTP, XML, UDDI, QoS y métrica.

2.1 Servicios Web

Existen múltiples definiciones sobre lo que son los Servicios Web, lo que muestra su complejidad a la hora de dar una adecuada definición que englobe todo lo que son e implican. Un WS es un conjunto de aplicaciones o de tecnologías con capacidad para interoperar en la Web. Estas aplicaciones o tecnologías intercambian datos entre sí con el objetivo de ofrecer servicios. Los proveedores ofrecen sus servicios como procedimientos remotos y los usuarios solicitan un servicio llamando a estos procedimientos a través de la Web [W3C, 2010].

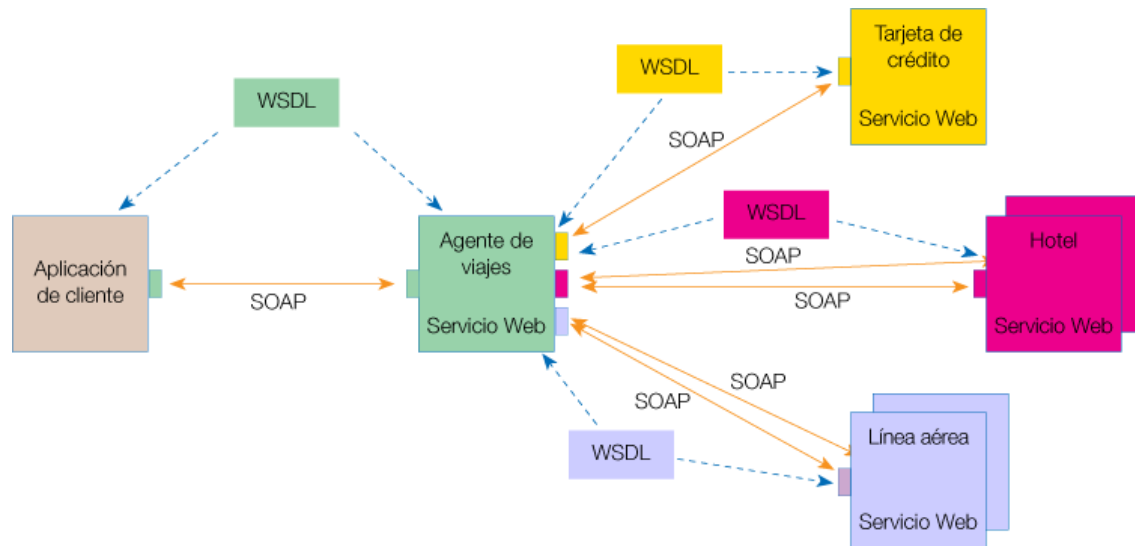


Fig. 1: Servicios Web en funcionamiento [W3C, 2010].

Para ejemplificar el uso de los WS se muestra el caso de la figura 1, un usuario, a través de una aplicación, solicita información sobre un viaje que desea realizar haciendo una petición a una agencia de viajes que ofrece sus servicios a través de Internet. La agencia de viajes ofrecerá a su cliente la información requerida. Para proporcionar la información, la agencia de viajes solicita a su vez información a otros recursos (otros Servicios Web) en relación con el hotel y la compañía aérea. La agencia de viajes obtendrá información de estos recursos, lo que la convierte a su vez en cliente de esos otros Servicios Web que le van a proporcionar la información solicitada sobre el hotel y la línea aérea. Por último, el usuario realizará el pago del viaje a través de la agencia de viajes que servirá de intermediario entre el usuario y el Servicio Web que gestionará el pago.

2.2 Protocolos Estándar de los Servicios Web

Los protocolos estándar para los Servicios Web se refieren a tecnologías tales como: SOAP, WSDL, HTTP, XML y UDDI que son la base tanto para construirlos, así como para hacer uso de ellos.

2.2.1 SOAP

La especificación Simple Object Acces Protocol (SOAP) define un marco de mensajería para el intercambio de datos con formato XML a través de Internet. El marco de mensajería es simple, fácil de desarrollar, y neutral por completo con respecto al sistema operativo, lenguaje de programación o plataforma de computación distribuida [Newcomer, 2002].

La W3C [W3C, 2004], define a SOAP como el conjunto formal de las convenciones que rigen las normas de formato y el procesamiento de un mensaje SOAP. Estos convenios incluyen las interacciones entre nodos SOAP, la generación y aceptación mensajes SOAP con el propósito de intercambiar información a lo largo de una ruta de un mensaje SOAP.

2.2.2 WSDL

El Lenguaje de Descripción de Servicios Web (WSDL) es un formato de esquema XML que define un marco extensible para describir las interfaces de los Servicios Web. WSDL fue desarrollado primeramente por Microsoft e IBM y fue enviado a W3C por 25 empresas. WSDL es el corazón del marco de Servicios Web, proporcionando una forma común para representar los tipos de datos que se pasan en los mensajes, las operaciones que se realizarán a partir de los mensajes, y el mapeo de los mensajes dentro de la red de transporte [Newcomer, 2002].

2.2.3 HTTP

Protocolo de transferencia de hipertexto (HTTP). Es un protocolo utilizado para la transferencia de datos a través de Internet, y que está basado en operaciones sencillas de solicitud y respuesta [W3C, 2008].

Los mensajes SOAP usan este protocolo como medio de transporte para el intercambio de mensajes entre servicios [W3C, 2004].

2.2.4 XML

XML es un Lenguaje de Etiquetado Extensible (Extensible Markup Language) muy simple, pero estricto, que juega un papel fundamental en el intercambio de una gran variedad de datos. Es un lenguaje muy similar a HTML pero su función principal es describir datos y no mostrarlos como es el caso de HTML. XML es un formato que permite la lectura de datos a través de diferentes aplicaciones [W3C, 2008].

La base fundamental sobre el cual los Servicios Web están construidos, provee un lenguaje para la definición de datos y de cómo procesarlos. XML representa una familia de

especificaciones relacionadas publicadas y sustentadas por la World Wide Web Consortium (W3C) y otras [Newcomer, 2002].

En el contexto de los Servicios Web, XML es usado no únicamente como el formato de mensaje sino que también como la forma en que los servicios son definidos. Por lo tanto, es importante saber acerca de XML, especialmente dentro del contexto de cómo es usado para definir e implementar los Servicios Web [Newcomer, 2002].

2.2.4.1 Propósito de XML

XML fue desarrollado para complementar el HTML y se utiliza especialmente para mejorar el soporte del manejo y creación de contenido dinámico. HTML es bueno para definir y mantener contenido estático, pero como la Web evoluciona hacia una plataforma de software, en los que los contenidos necesitan ser generados y digeridos dinámicamente. El uso de XML, puede definir cualquier número de los elementos que se asocian con el significado de los datos, es decir, que describen los datos y qué hacer con ellos, utilizando uno o más elementos creados para el propósito, por ejemplo:

```
<Empresa>
  <NombreEmpresa region="US">Skateboot Manufacturing</NombreEmpresa>
  <Dirección>
    <linea>200 Hig Street</linea>
    <linea>Springfield, MA 55555</linea>
    <Pais>USA</Pais>
  </Dirección>
  <Teléfono>+1 781 555 5000</Teléfono>
</Empresa>
```

En el ejemplo anterior, XML permite definir no únicamente elementos que describen el dato sino que también estructuras de ese grupo de datos relacionados. Es fácil imaginar una búsqueda de elementos que cumplen ciertos criterios, tales como <País> y <Teléfono> para una determinada empresa, o para todos elementos <Empresa> y para retornar una lista de estas entidades identificándose ellas mismas como empresas en la Web [Newcomer, 2002].

2.2.5 UDDI

La W3Schools [W3Schools, 2013] define a la UDDI como una estructura independiente de la plataforma para la descripción de los servicios, el descubrimiento de las empresas, y la integración de servicios de negocio a través de Internet.

Después de haber definido los datos en los mensajes (XML) y descrito los servicios que van a recibir y procesar el mensaje (WSDL) e identificado los medios de envío y recepción de los mensajes (SOAP), se necesita una forma para publicar el servicio que se ofrece y buscar los

servicios que otros ofrecen y que se puedan usar. Esta es la función que UDDI (Universal Distribution, Discovery, and Integration) proporciona [Newcomer, 2002].

2.3 QoS (Calidad de Servicio)

De acuerdo con la norma ISO 8402, la palabra calidad es definida como “la totalidad de características de una entidad que tiene la habilidad para satisfacer necesidades expresadas o implícitas”. En la ISO 9000 se define calidad como el grado para que un conjunto inherente de características cumplan requerimientos. Básicamente la [International Telecommunication Union] (Recomendación E.800 [ITU-TE.800]) y ETSI [ETSI-ETR003] define Calidad de Servicio (QoS) como “el efecto colectivo del desempeño del servicio con determinado grado de satisfacción de un usuario del servicio” [Marchese, 2007].

2.4 Medición, Medida y Métrica

- Medición: es el acto de determinar una medida [Pressman, 2002].
- Medida: proporciona una indicación cuantitativa de la extensión, cantidad, dimensiones, capacidad o tamaño de algunos atributos de un proceso o producto [Pressman 2002].
- Métrica: según el “IEEE Standard Glossary of Software Engineering Terminology” [IEEE, 1990] se define al término métrica como: “una medida cuantitativa del grado en que un sistema, componente o proceso posee un atributo dado” .

Los conceptos antes descritos en este capítulo son utilizados para el desarrollo de esta tesis y que servirán de base al lector para su buen entendimiento. Estos conceptos están directamente relacionados con los Servicios Web, los cuales, se basan en la Arquitectura Orientada a Servicios (SOA).

2.5 Definiciones de Cloud y Grid Computing

Las definiciones que se mencionan a continuación, son consideradas en su mayoría, como definiciones de consenso.

“La nube es una gran fuente de recursos virtualizados fácilmente usables y accesibles (tales como hardware, plataformas de desarrollo o servicios). Estos recursos pueden reconfigurarse dinámicamente para adaptarse a una carga variable (escala), lo que también permite una utilización óptima de los recursos” [Patidar, 2012] .

“Cloud Computing es un modelo que permite, cómodo acceso a la red bajo demanda en todas partes a un conjunto compartido de recursos informáticos configurables (por ejemplo, redes, servidores, almacenamiento, aplicaciones y servicios) que pueden ser rápidamente suministrados y liberados con un mínimo esfuerzo de gestión rápida o interacción con el proveedor del servicio” [Hashemi, 2012].

Otras definiciones similares se presentan en los trabajos de los siguientes autores: [Mell, 2011], [Zhang, 2010], [Wang, 2010], [Vaquero, 2009] y [Foster, 2008].

2.6 Algunas definiciones de “Grid Computing”

“Grid Computing es una forma de computación distribuida que implica la coordinación y el intercambio de computación, aplicaciones, datos, almacenamiento y recursos de red a través de organización dinámica y geográficamente dispersos” [Hashemi, 2012].

“Un sistema que coordina recursos que no están sujetos a un control centralizado, usando protocolos e interfaces estándar, abiertos y de propósito general para entregar características de servicio no triviales” [Vaquero 2009].

Las definiciones de Cloud Computing aquí citadas son consideradas, en su mayoría, como definiciones de consenso, así como también se remarca que no hay una definición estándar aún. En los trabajos de [Zhang 2010], [Hashemi 2012] toman la definición del Instituto Nacional de Estandares y Tecnología (NIST por sus siglas en ingles) de los EE.UU [Mell 2011], debido a que cubre todos los aspectos esenciales de Cloud Computing. La intención de presentar este listado de definiciones es para identificar en tales definiciones algunas de las características de cada uno de estos dos paradigmas.

CAPÍTULO 3: Trabajos Relacionados

En este capítulo se describen los trabajos relacionados encontrados en la literatura actual que tratan la importancia de los atributos de calidad en lo que respecta a Servicios Web.

3.1 Trabajos relacionados

En este apartado se describen los trabajos que proponen el uso de atributos de calidad en los perfiles de los Servicios Web para propósitos de selección, composición o reutilización. Los trabajos relacionados, a diferencia de los trabajos del estado del arte que se discuten en el capítulo 4, los autores se enfocan en resaltar la importancia de incluir atributos de calidad de servicio en los perfiles de WS.

3.1.1 Propuestas para incluir atributos de calidad en los WS

Es esencial incluir los atributos de calidad en los Servicios Web para dar lugar a la selección y composición de WS con respecto a requerimientos no funcionales. En el reporte técnico de la W3C se proponen algunos requerimientos no funcionales para aspectos de calidad de Servicios Web, así como también se mencionan algunos enfoques, como por ejemplo, la extensión de la UDDI para soporte de QoS y una extensión de SOAP para encontrar Servicios Web que satisfagan requerimientos de atributos de calidad específicos [W3C, 2003].

La Computación Orientada a Servicios (SOC), se describe como algo que está compuesta por las siguientes capas: servicios básicos, composición de servicios y administración de servicios, estas capas en conjunto describen a una arquitectura orientada a servicios extendida. Dentro de la capa de servicios básicos se dan las acciones de publicación, descubrimiento, selección y enlace de un WS; así como también, su calidad, comportamiento, interfaz y capacidad que son de interés tanto del proveedor como del cliente. Dentro de la capa de composición se encuentran funciones correspondientes al servicio compuesto como lo son: coordinación entre la ejecución de los componentes del servicio, monitoreo, conformidad y la calidad del servicio compuesto. La capa de administración de servicio de SOA es para proporcionar soporte a negocios electrónicos como servicios con valor agregado [Papazoglou, 2003].

Un estudio de seis enfoques diferentes en [Tian, 2007] hacia la especificación y gestión de WS compatibles con QoS, los seis enfoques que se mencionan son: “*Web Service Level Agreement*”, “*Web Service Offering Language*”, “*SLang*”, “*UX*”, “*UDDIe*”, “*WS-QoS*”, debido a los diferentes aspectos que tratan, según el autor, resulta difícil compararlos con respecto a un criterio coherente.

La relación entre SOA y los atributos de calidad tiene un impacto en el logro de los objetivos de negocios en las empresas que deciden implementar o diseñar un sistema con enfoque SOA [O'Brien, 2005]. Los atributos de calidad son necesarios para los objetivos específicos de negocios, por ejemplo, los objetivos de negocio como agilidad y ser el primero en el mercado puede significar que la adaptabilidad, extensibilidad, escalabilidad e interoperabilidad se convierten

en los principales motores para la arquitectura de la aplicación y el desarrollo de los WS. Por otra parte el autor [Nurhayati, 2008] menciona que, los atributos de calidad no únicamente deberían ser medidos sino que también previstos durante las etapas de desarrollo e implementación, sin embargo, hay retos y limitantes para determinar y elegir requerimientos de QoS para alta calidad de WS. El autor introduce un metamodelo de QoS en las etapas de desarrollo e implementación para la predicción de QoS, además de argumentar que existe poco trabajo disponible sobre predicción de calidad de WS.

A diferencia de los trabajos anteriores que sólo remarcen la necesidad de incluir QoS en los Servicios Web, el autor Zeng discute el cálculo de métricas de atributos de calidad de WS mediante monitoreo de ejecuciones a través de una infraestructura SOA para permitir la detección semántica y enrutamiento de eventos operacionales de WS. Se describe la implementación un motor de alto rendimiento para el cálculo de métrica que puede soportar el desempeño más alto de eventos y se propone un modelo de observación que permite a los usuarios definir el tipo de métricas y fórmulas [Zeng, 2007].

Sin duda alguna, incluir atributos de calidad en los perfiles de los WS favorece la selección y gestión de WS, pero, para lograr esto, algunos autores proponen la extensión de la UDDI para soportar tales descripciones o extender SOAP para encontrar WS compatibles con atributos de calidad, así como también, favorece al monitoreo de estos. SOA permite proporcionar soporte a negocios electrónicos como servicios con valor agregado y ayuda en el logro de los objetivos de negocio.

3.1.2 Selección de Servicios Web

Los atributos de calidad son la clave para dar lugar a la selección dinámica de servicios. Una clasificación de atributos de calidad se menciona en [Maximilien, 2004], la cual se divide en objetivos: confiabilidad, disponibilidad y solicitud para el tiempo de respuesta o subjetivos: centrándose en la experiencia del usuario. La selección dinámica de WS a través de un marco de agente, junto con una ontología de calidad de servicio.

Un enfoque en [Tian, 2004] es propuesto el cual permite una selección eficiente, dinámica, compatible con QoS y monitoreo de WS. El prototipo de este enfoque es implementado con la tecnología .NET que está compuesto por: un editor WS-QoS para la especificación de propiedades QoS, un administrador de requerimientos de WS-QoS para recuperar requerimientos QoS especificados por aplicaciones cliente, un agente de Servicio Web para una selección eficiente compatible con QoS de los Servicios Web ofertados y un Monitor de WS-QoS para verificar el cumplimiento de los servicios ofertados.

A diferencia de los trabajos anteriores, en una selección dinámica de WS pero a través de requerimientos funcionales, se utiliza un agente, cuando el agente encuentra un conjunto de servicios que satisfacen tales requerimientos, entonces los requerimientos pueden ser usados como criterio de valoración del servicio, posteriormente, para refinar la selección, se requiere de un método de selección, este método incluye una función de agregación de criterio múltiple que combina métricas LSP (Logic Scoring Preference) con operadores OWA (Ordered Weighted Averaging) [Yu, 2008].

Agentes pueden ser usados para la selección de servicios tanto para seleccionar a aquellos que son compatibles con QoS o no. La selección de servicios puede ser de forma dinámica, eficiente, compatible con QoS y monitoreo. El monitoreo de QoS es importante pues a través de un Monitor o técnicas de monitoreo de QoS se puede verificar el cumplimiento de los servicios ofertados. Los atributos de calidad pueden ser agrupados o clasificados en subjetivos y objetivos.

3.1.3 Composición de Servicios Web

Con el fin de facilitar y automatizar la composición de Servicios Web, algunos enfoques sobre composición de servicios se describen a continuación. El uso de la web semántica y tecnologías de razonamiento automatizado es usado para describir, simular, componer, probar y verificar la composición de WS. Para describir las capacidades de los WS se toma a la ontología DAML-S DAML+OIL, se define la semántica para un subconjunto relevante de DAML-S en términos de un lenguaje lógico de primer orden. Con la semántica, se codifica la descripción del servicio mediante un formalismo de red de Petri que proporciona procedimientos de decisión para la simulación, verificación y composición de Servicios Web [Narayanan, 2002].

Un mecanismo que permite verificar si un conjunto de servicios seleccionados para composición satisface los requerimientos de calidad para toda la composición a través de patrones de composición abstracta como lo son: secuencia, ciclo o ejecución en paralelo es propuesto en [Jaeger, 2004]. En el trabajo de [Mounla, 2008] se emplea un enfoque similar al de [Narayanan, 2002] pues se usa razonamiento basado en casos junto con la programación entera, que es un subconjunto de la programación lineal para la composición compleja de WS.

Un enfoque de composición de WS “*end to end*” se discute en [Agarwal, 2005] para ello divide la composición en dos fases: composición lógica y composición física, en la composición lógica proporciona nuevas funcionalidades en el servicio que antes no estaban disponibles, mientras que en la composición física permite la selección de las instancias de componentes de servicio basado en requerimientos no funcionales que luego se unirán entre sí para desplegar el

Servicio Web compuesto recién creado. Por último tenemos a [Frankova, 2005] el cual hace una comparación de enfoques de modelado de la calidad de composición de WS, los enfoques estudiados son los siguientes: “*Markov Chains*”, “*Quality Vector*”, “*Agent-oriented*”, “*Ontology-based*” los cuales son comparados contra los siguientes criterios “*Objective QoS*”, “*Subjective QoS*”, “*Run-time support*”, “*QoS assignment*”, “*Requirements considering level*”.

Diferentes enfoques pueden ser usados dentro de la composición de servicios compatibles con QoS, dándose esta en flujos de composición de WS.

3.1.4 Composición de Servicios Web y procesos de negocio

Los WS pueden ser utilizados como implementación de actividades dentro de un proceso de negocios y que estos puedan ser externalizados [Leymann, 2002]. Otros puntos importantes dentro de la composición es la orquestación y coreografía, en este caso, la orquestación se refiere a un proceso de negocio ejecutable que puede interactuar tanto con Servicios Web internos como externos. Mientras que la coreografía realiza un seguimiento de la secuencia de mensajes entre Servicios Web. Los estándares usados tanto para la coreografía y orquestación son los siguientes: BPEL4WS, WSCI y BPML [Peltz, 2003].

3.1.5 Reutilización de Servicios Web

La reutilización de servicios puede darse de diferentes maneras; por ejemplo, al desarrollar servicios con el enfoque SOA este proporciona grandes beneficios, como lo es la reutilización de Servicios Web existentes dentro de una empresa cuando se desea crear una aplicación con nuevas características. Algunos otros autores presentan enfoques que permiten almacenar a servicios compuestos en la UDDI u otros repositorios definidos por el autor para su uso futuro.

La composición del Servicio Web se divide en dos fases: composición lógica y composición física. Los desarrolladores de Servicios Web pueden mantener un registro de WS que va más allá de la tradicional UDDI mediante la incorporación de descripciones semánticas de los componentes. Cuando un requerimiento nuevo de servicio surge, se puede expresar en el contexto de una ontología de dominio. El entorno de creación de servicio se puede utilizar para generar flujos de trabajo posibles para lograr la funcionalidad deseada reutilizando WS existentes, para tal caso, una herramienta es propuesta para la composición de servicios. Estos servicios son servicios compuestos que son creados dentro de la etapa de composición lógica, los cuales son almacenados en el registro de capacidades de servicio [Agarwal, 2005].

Cuando las limitaciones de calidad y las preferencias son asignadas a los servicios compuestos en lugar de los servicios componentes, es fácil volver a utilizar el servicio compuesto,

en los enfoques “*Quality Vector*” y “*Agent-oriented*”, que se mencionan, pero que no se describen [Frankova, 2005]. Con respecto al enfoque SOA, en [O'Brien, 2005], los autores argumentan que los WS existentes dentro de una organización, estos pueden ser reutilizados en la construcción de una aplicación. Servicios proporcionados desde fuera de la organización pueden ser también considerados cuando se construye una aplicación. La capacidad de reutilizar los servicios internos y externos dentro de un enfoque SOA, combinar servicios de formas nuevas y diferentes y añadir servicios adicionales donde sean necesarios reduce el tiempo de desarrollo de nuevas aplicaciones y llevarlas al mercado, aunque no se menciona alguna implementación de reutilización.

La reutilización del WS se da cuando la composición de WS es optimizada globalmente usando programación entera y esta es almacenada en un repositorio de razonamiento basado en casos para uso futuro, lo cual es un enfoque híbrido, este es presentado en [Mounla, 2008].

En las tablas 1 y 2 se presentan trabajos de la literatura que tratan la importancia de incluir los atributos de calidad en la selección, composición o reutilización de un Servicio Web. En comparación con los trabajos antes mencionados, este trabajo de investigación no sólo plantea la importancia del uso de atributos de calidad en los Servicios, sino que además, presenta un estudio de estos atributos así como la forma de medirlos, para ello se presenta una taxonomía que describe el análisis de trabajos encontrados y una clasificación de atributos de calidad para WS.

Tabla 1: Tabla comparativa de trabajos relacionados.

Artículo	Dominio	Objeto de estudio	Atributos de calidad	Taxonomía de análisis de enfoques para el tratamiento de QoS	Clasificación de Atributos de Calidad
[Narayanan, 2002]	Composición	QoS	No especificados	No	No
[Leymann, 2002]	Composición	Procesos de negocio	No especificados	No	No
[W3C, 2003]	Propuesta de uso de QoS	QoS	Performance, Reliability, Scalability, Capacity, Robustness, Exception handling, Accuracy, Integrity, Accessibility, Availability, Interoperability, Security	No	No
[Peltz, 2003]	Composición	Procesos de negocio	No especificados	No	No
[Papazoglou, 2003]	Propuesta de uso de QoS	QoS	Cost, performance, security, authentication, privacy, (transactional) integrity, reliability, scalability, and availability.	No	No
[Tian, 2004]	Selección	QoS	No especificados	No	No
[Maximilien, 2004]	Selección	QoS	Confiabilidad, disponibilidad y solicitud para el tiempo de respuesta	No	si
[Tian, 2007]	Propuesta de uso de QoS	QoS	No especificados	No	No
[Jaeger, 2004]	Composición	QoS	No especificados	No	No

Tabla 2: Tabla comparativa de trabajos relacionados (continuación).

Artículo	Dominio	Objeto de estudio	Atributos de calidad	Taxonomía de análisis de enfoques para el tratamiento de QoS	Clasificación de Atributos de Calidad
[Agarwal, 2005]	Composición, reutilización	QoS	No especificados	No	No
[O'Brien, 2005]	Propuesta de uso de QoS, Reutilización	QoS	Interoperability, Reliability, Availability, Usability, Security, Performance, Scalability, Extensibility, Adaptability, Testability, Auditability, Operability and Deployability, Modifiability	No	No
[Zeng, 2007]	Propuesta de uso de QoS	QoS	No especificados	No	No
[Mounla, 2008]	Composición, reutilización	QoS	No especificados	No	No
[Nurhayati, 2008]	Propuesta de uso de QoS	QoS	Reliability, Execution Price, Availability, Accessibility, Capacity, Performance, Security, Transaction, Integrity, Regulatory, Reputation.	No	No
[Yu, 2008]	Selección	Requerimientos funcionales	No especificados	No	No
[Frankova, 2005]	Composición, reutilización	QoS	No especificados	No	No
Trabajo que se reporta en este documento	Reutilización, selección y composición	QoS y sus métricas	Los QoS encontrados en la literatura actual	Si	Si

CAPÍTULO 4: Presentación del estado del arte

En este capítulo se discuten los trabajos seleccionados en la literatura que plantean la medición de atributos de calidad de Servicios Web dentro de los rubros de selección, composición y reutilización. También se presenta una vista del análisis de los trabajos encontrados. Adicionalmente se describen las propuestas del grupo de trabajo de la W3C, así como QoS en la nube y un modelo de calidad de WS propuesto por OASIS.

Los trabajos discutidos en este capítulo se enfocan principalmente, en cómo pueden ser medidos los atributos de calidad de Servicios Web, es importante también señalar que según los trabajos analizados no todos los atributos cuentan con una métrica para su medición, en tal caso, únicamente existe su definición o concepto.

La taxonomía de la figura 2, presenta un esquema general de los puntos a tratar en este estudio de atributos de calidad de Servicios Web, los cuales son centrales de este capítulo. En cada uno de los dominios de selección, composición y reutilización de WS, se realiza una discusión sobre lo encontrado en la literatura acerca de atributos de QoS y sus respectivas métricas en caso de que existan. Los atributos y métricas encontradas son presentados en las secciones 4.7.2 y 4.7.3.

Este trabajo también explora los atributos de calidad dentro de la nube computacional, así como también se menciona la información generada por el grupo de trabajo de la W3C que tiene que ver directamente con los Servicios Web, ya que la W3C es la que norma a los WS. Así mismo comprende el modelo calidad de Servicios Web propuesto por OASIS llamado WSQM (Web Service Quality Model).

Dentro de los trabajos estudiados, en [Papaioannou, 2006] se presenta un lenguaje ontológico de QoS, el cual ayuda a describir parámetros de atributos de calidad arbitrarios, además de enfatizar la importancia de incluir atributos de calidad en los WS, pues incluir atributos de calidad en los perfiles de WS da lugar a la selección y composición automática de WS de acuerdo a requerimientos no funcionales de los usuarios. El lenguaje ontológico de QoS mencionado en el trabajo de Papaioannou consta de un conjunto de reglas, clases, propiedades de objeto, propiedades de datos que son usados para representar todos los parámetros heterogéneos de QoS junto con las relaciones entre ellos.

El lenguaje ontológico de QoS permite definir las relaciones entre atributos de calidad y la forma en que se miden. Aunque en esta tesis de investigación no se realiza un estudio conforme a las relaciones de las clases como se describe en tal lenguaje ontológico, el trabajo de [Papaioannou, 2006] fue útil para abordar el estudio de los atributos de calidad, ya que nos proporciona la pauta de que los atributos de calidad son importantes en los perfiles de WS para dar lugar a la selección y composición y que tales atributos pueden ser medidos, así como las clases que son parte de la ontología, nos dan la idea de qué características debe tener un atributo de calidad.

4.1 Taxonomía

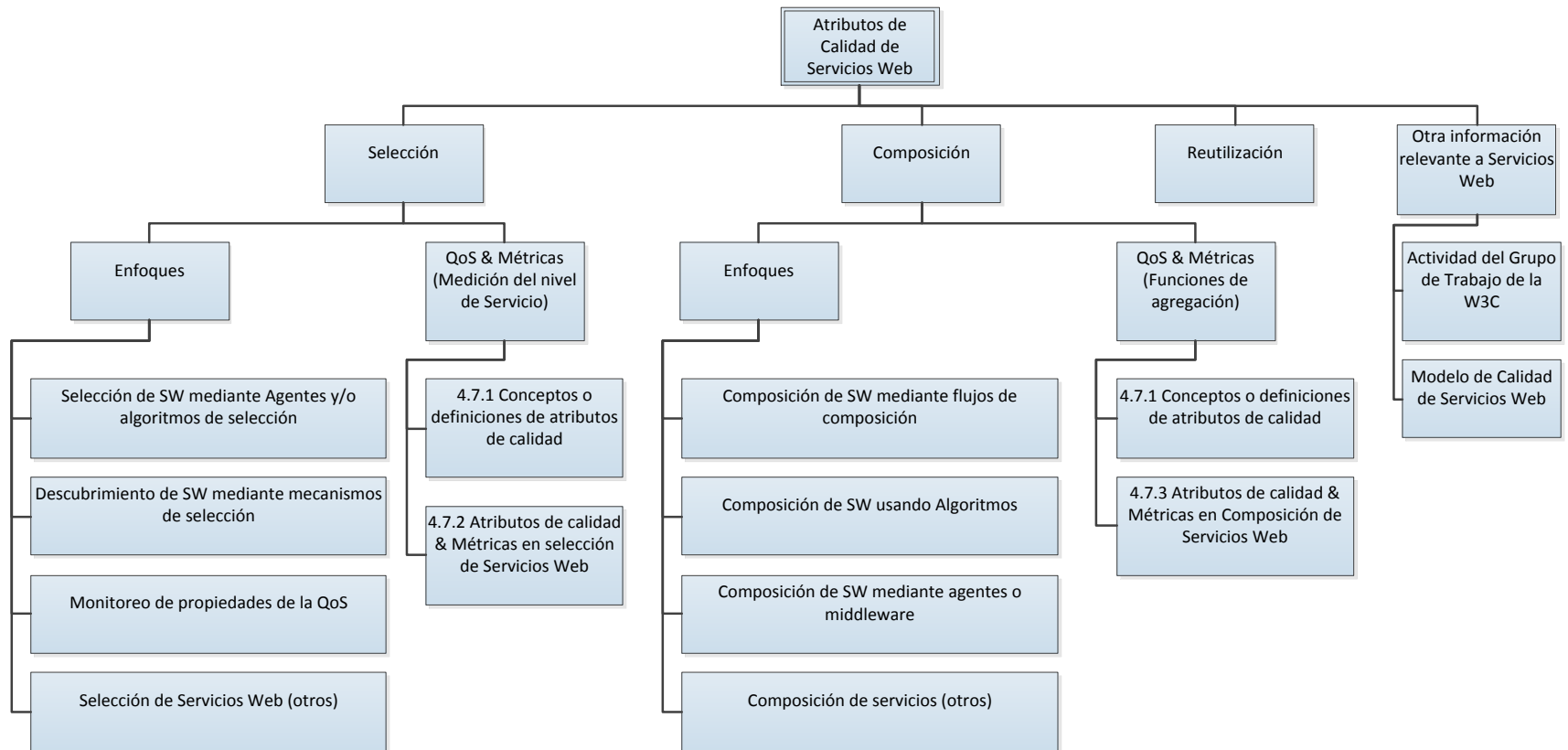


Fig. 2: Taxonomía de atributos de calidad de WS.

Otros puntos o criterios importantes tomados en cuenta para atender el estudio de atributos de QoS y que son de aportación en este trabajo, se mencionan a continuación:

- **Su definición:** La definición o concepto del atributo de calidad.
- **Su objetivo:** Que es lo que se pretende medir o se está midiendo, como por ejemplo, tiempo de respuesta del servicio, la capacidad del servicio en responder peticiones simultáneas, la robustez, etc.
- **Formas de medición:** Son las diferentes métricas que existen para medir un atributo de calidad.
- **En que dominio o contexto pueden ser aplicados:** Los atributos de calidad podrían agruparse por la similitud entre estos, es decir, si se encuentran un mismo contexto o dominio. Esto permitiría entonces tener conocimiento para resolver problemas en ciertos contextos.

4.2 Selección de WS

4.2.1 Selección de WS mediante Agentes y/o algoritmos de selección

Con el fin de ofrecer o permitir el manejo de calidad en la selección de WS se han propuesto marcos de trabajo que permitan evaluar los atributos de calidad de un gran número de Servicios Web. En [Patel, 2003] se diseña un modelo de QoS para la selección, enlace y ejecución de WS, así como también se desarrolla un conjunto de algoritmos para calcular los parámetros de QoS y la implementación de ellos usando un sistema basado en reglas; en [Liu, 2004] además de realizar el cálculo de QoS para la selección de WS, se presenta un registro de QoS, es decir, una BD en donde guardar los QoS de los usuarios y la valoración del servicio. Para el registro de QoS se construyó un mecanismo de vigilancia el cual exige un usuario y contraseña para actualizar el registro de QoS.

Poder seleccionar WS de manera automática es de mucha utilidad para el usuario, ya que, el seleccionar servicios en tiempo de diseño resulta hacer una elección aleatoria, que es una elección ciega, por lo tanto proporcionar mecanismos de selección automática, son necesarios para ayudar a consumidores a distinguir servicios buenos de los malos. Tanto en [Yu, 2005] y [Thio, 2005], se utiliza un agente que se encarga de seleccionar y coordinar el componente de servicio individual, se diseñan algoritmos de selección de servicios usados por agentes de QoS para construir el servicio compuesto óptimo. En la selección de componentes de servicio se modela como el problema de “*Knapsack*” de opciones múltiples y el enfoque gráfico (teoría de grafos).

QoS en los Servicios Web es una característica importante para que los agentes puedan cumplir con su tarea de selección o composición de WS como se menciona en [Badr, 2008],[De Cock, 2007],[Lin, 2008],[Rajendran, 2010] y [Taher, 2005]. Para realizar las tareas de selección, el usuario puede especificar sus requerimientos tanto funcionales como no funcionales; las propiedades funcionales de un WS son las cosas que puede hacer y las propiedades no funcionales muestran como el servicio puede hacerlo [Badr, 2008]. En [Taher, 2005] se presenta un marco que tiene dos partes: un modelo de datos y otro de cálculo. El modelo de datos proporciona una ontología para proporcionar la comprensión común de los atributos de calidad y su semántica, el modelo de cálculo usa una medida de distancia de similitud, para la selección de servicios y permite la gestión de los cambios dinámicos de las propiedades de QoS. La especificación de los requerimientos de QoS puede ser realizada de forma estricta o flexible; usuarios que especifican sus requerimientos no funcionales de forma estricta tienen el riesgo de no encontrar ningún servicio que satisfaga sus requerimientos, para tal caso se requiere de un algoritmo de selección de WS que permita cierta flexibilidad en la especificación de QoS [De Cock, 2007]. Un agente basado en SOA es propuesto en [Rajendran, 2010] para facilitar la selección dinámica de WS a través de requerimientos tanto funcionales y no funcionales del consumidor.

Uno de los problemas en selección de WS son los distintos puntos de vista sobre la calidad de servicio tanto de los proveedores y los consumidores. El proveedor, por ejemplo, considera confiable su servicio si su tasa de éxito es del 90%, mientras que para el cliente es confiable si la tasa de éxito es del 99%, para tal caso, en [Lin, 2008] se propone resolver este conflicto mediante un consenso entre el cliente y el proveedor de servicios, el cual consta de un enfoque de moderación de consenso de QoS. El consenso “QoS Consensus Moderation Approach” (QCMA) trata de resolver este conflicto, de manera que, tanto los usuarios y proveedores de servicios puedan expresar sus requerimientos de QoS usando términos fuzzy, tales como, “muy confiable” y “poco confiable”.

Tanto en selección como en composición de WS se usan agentes para facilitar la tarea de la selección del Servicio Web idóneo para el usuario dentro de un conjunto de servicios. Debido al creciente número de WS suceden dos cosas: la primera es que hay más opciones de donde escoger, pero, en segundo lugar, tenemos que esto se vuelve un problema de optimización en composición de servicios cuando el número de servicios existentes es grande. El agente de selección de servicios debe ser capaz de optimizar tanto el tiempo de selección y el subconjunto de servicios entregados deben ser los mejores u óptimos y que coincidan con los requerimientos tanto funcionales como no funcionales del usuario.

4.2.2 Descubrimiento de WS mediante mecanismos de selección

Un mecanismo de descubrimiento es el proceso de encontrar un proveedor de servicio apropiado para un solicitante de servicio a través de un agente intermedio [Garofalakis, 2004], básicamente el descubrimiento de servicios tiene los siguientes pasos: 1) El proveedor de servicio publica sus capacidades a los agentes intermedios. 2) Agentes intermediarios almacenan esta información. 3) Un solicitante de servicio le solicita a un agente intermedio si sabe de prestadores de servicios que mejor coincidan con las capacidades solicitadas. 4) El agente intermedio, con el fin de responder, intenta hacer coincidir la solicitud con los registros almacenados y devuelve un subconjunto de registros de los proveedores de servicios almacenados. Un mecanismo se apoya de las descripciones de QoS en WS ya sea a través de la extensión del WSDL o de la UDDI para soportar tal extensión de descripciones de QoS llamadas políticas de QoS o WS-Policy [Garcia, 2006]. En [Rajendran, 2009] sucede algo similar, se propone un enfoque tModel para resolver el problema de almacenamiento de QoS en la UDDI. El autor [Ran, 2003], propone extender la UDDI para incluir descripciones de QoS.

Un mecanismo es usado para verificar la información de calidad localizada dentro del archivo de descripción del WS localizado en la UDDI o en una URL independiente, además de implementar una BD para almacenar información de QoS para WS. Para la selección de WS se usa un módulo, el cual maximiza el valor de los atributos de calidad entre otros con la misma funcionalidad [Diamadopoulou, 2008]. Según Diamadopoulou, la medición de QoS de un WS debería de corresponder a lo siguiente: ¿Qué medir? ¿Cómo medir? ¿Quién hace la medición? Y ¿Dónde se toman las medidas? Por otra parte menciona que el grupo de características QoS que son más importantes para sistemas distribuidos aún no ha sido determinado, aunque se cree que las características o atributos de calidad más importantes de un servicio son: “*availability*”, “*security*”, “*throughput*” y “*response time*”. En [Wang, 2007], el mecanismo que usa para la selección de WS, se centra en la confianza y reputación del WS.

Un motor de búsqueda para el descubrimiento de WS se presenta en [Vu, 2006] quien propone un enfoque para la descripción semántica y descubrimiento de WS tomando en cuenta especialmente sus atributos de calidad, el motor de búsqueda está basado en un modelo de descubrimiento algebraico, el motor puede ejecutarse como un servicio centralizado para entornos de pequeña escala. Un mecanismo se apoya de Agentes, así como de la UDDI, como por ejemplo, la UDDI en su versión 3, diversos autores proponen extenderla de manera que esta permita especificar descripciones de QoS de WS, todo esto con el objetivo de facilitar el descubrimiento de WS.

4.2.3 Monitoreo de propiedades de QoS

En un proceso dentro de un sistema distribuido cuando el cliente invoca los objetos remotos del servidor, el servidor recibe las peticiones entrantes, las procesa y devuelve la respuesta; las propiedades de calidad relacionadas con “performance” y negociadas dentro de un Acuerdo del Nivel del Servicio pueden ser medidas, según los autores [Oberortner, 2011],[Oberortner, 2010],[Rajendran, 2009] y [Rosenberg, 2006] argumentan que las propiedades relacionadas con “performance” son las siguientes: *Round-Trip Time*, *Marshaling Time*, *Network Latency*, *Response Time*, *Processing Time*, *Unmarshaling Time*, *Execution Time*. Para realizar las mediciones de QoS relacionadas con “performance” se pueden utilizar patrones de medición tanto del lado del cliente o del servidor: *QOS INLINE*, *QOS WRAPPER*, *QOS INTERCEPTOR*, *QOS REMOTE PROXY* e *INDERECTION*. Es importante destacar que si las mediciones tanto del cliente como del servidor se realizan de forma independiente, ambos pueden falsear los resultados.

En [Rosenberg, 2006] además de las métricas relacionadas con “performance” el autor menciona tres atributos relacionados con “*Dependability*” los define como *availability*, *accuracy* y *robustness*. Otros trabajos como: [O'Brien, 2005], [Ran, 2003] y [Yu, 2007] mencionan propiedades o atributos de QoS relacionados con el desempeño.

Con respecto al desempeño del servicio en la nube, [Xiong, 2009] calcula el percentil del tiempo de respuesta y es presentado al cliente como un estadístico. El percentil del tiempo de respuesta es hallado mediante la función de distribución de probabilidad del tiempo de respuesta $T(t)$ y se deriva un método de aproximación usando la transformada de Laplace para el cálculo de la probabilidad y distribuciones acumuladas del tiempo de respuesta.

A diferencia de los trabajos anteriores que mencionan atributos de calidad relacionados con “performance”, en el trabajo de [Jurca, 2007] se realiza un monitoreo de QoS basado en la opinión de calidad de los clientes. Se propone un mecanismo de reputación que recoge las calificaciones o valoraciones de los clientes y calcula la calidad real entregada del servicio. Para este trabajo se usa la métrica “*verity*”, esta métrica toma en cuenta tanto la reputación de un servicio así como los términos del Acuerdo del Nivel del Servicio.

El monitoreo de propiedades de QoS de los WS tiene como objetivo el vigilar el cumplimiento de la calidad entregada por el proveedor y tomar medidas que prevengan la violación de los acuerdos de Servicio (SLA).

4.2.4 Selección de Servicios Web en otros dominios

Debido al aumento de WS con funcionalidades similares, los atributos de calidad vienen a ser un criterio esencial de selección. En [Chaari, 2008] se propone un marco de trabajo, que propone una categorización ontológica de las propiedades no funcionales, los atributos de calidad son presentados como políticas y agregados a la descripción del WS, estas políticas son usadas por un motor de selección el cual se encarga de seleccionar y comparar los servicios mediante funciones de desigualdad entre políticas.

Comúnmente los atributos de calidad más usados para la selección de servicios son: “*availability*”, “*security*”, “*throughput*” y “*response time*” entre otros. El autor [Kalepu, 2003] propone un atributo de calidad llamado “*verity*” y argumenta que tal atributo debería ser tomado en cuenta para una selección y composición de Servicios Web conducida por calidad.

La medición del nivel del servicio es importante tanto para el cliente y el proveedor de servicios. El autor [Lee, 2010] divide en dos grupos los factores de calidad: factores de calidad de los valores de negocio y factores para la medición del nivel de servicio. Los primeros, resultan difíciles de medir o evaluarlos objetivamente, debido a que son valores subjetivos y por lo tanto, dependen de la opinión de los usuarios de WS, mientras que los factores de calidad de medición del nivel del servicio pueden ser medidos sistemáticamente y objetivamente, aunque no se proporcionan detalles de medición; los factores son los mismos que propone [OASIS, 2010].

Evaluar atributos de calidad de WS del mundo real resulta difícil a gran escala desde localizaciones distribuidas, debido a que las invocaciones consumen recursos de los proveedores de servicio e imponen costos para los usuarios. Además, es difícil recoger datos de atributos de calidad desde los usuarios de los servicios distribuidos según [Zheng, 2010].

De manera diferente a los trabajos anteriores, en el trabajo de [Hwang, 2007] se propone un modelo de QoS basado en la probabilidad para Servicios Web atómicos y compuestos, el cual considera la medición de un atributo de calidad como una variable aleatoria discreta con una función de probabilidad de masa, además de contar con un marco computacional para derivar medidas de QoS de un flujo de trabajo de un Servicio Web, con la finalidad de realizar estimaciones para la elección de componentes de Servicios Web entre un conjunto de candidatos de WS.

Con respecto a la administración de WS en el trabajo de [Liu, 2010] se aborda la importancia de la gestión de estos, con el fin de proporcionar WS de alta calidad, para ello los autores diseñan e implementan un marco para la administración dinámica de Servicios Web basada en QoS en conjunto con un modelo extensible de QoS, es decir, al modelo se le pueden

agregar más atributos de calidad. El marco es implementado en el sistema “*web services-based remote sensing data-sharing system*”.

Una investigación se presenta sobre los WS en la WWW con base a datos estadísticos que determinan el estatus de los WS. Los resultados indican que el 63% de los WS disponibles sobre la Web son considerados como activos. Típicamente un WS es publicado en un repositorio UDDI, aunque no todos los repositorios están dentro de un estándar, aquellos que no toman en cuenta un estándar tienden a ser vistos como inseguros y han desaparecido, por ejemplo, “*BindingPoint*”, “*Woogle*”, and “*SalCentral*”. Debido a lo anterior, se propone un motor de rastreo específico que se pueda utilizar para el descubrimiento de WS que se ajuste a una arquitectura de WS apropiada [Al-Masri, 2008].

En todos los trabajos discutidos en esta sección, se da la importancia a la inclusión de atributos de calidad en los perfiles de WS tanto para dar lugar a la selección y composición, así como también para la gestión de WS aunque, la interpretación del valor de los atributos de calidad de los consumidores y proveedores del WS puede ser muy diferente, es aquí donde toma importancia el tema de consenso sobre QoS propuesto por el autor [Lin, 2008].

4.2.5 SLA

El SLA o ANS (Acuerdo de Nivel de Servicio) es importante para los Servicios Web pues es un acuerdo entre el proveedor y el cliente sobre el nivel de servicio que se entrega o recibe el cliente y que el proveedor está obligado a cumplir.

Un SLA es usado para llegar a un consenso en términos del nivel de calidad del servicio, por ejemplo: tiempo de respuesta, disponibilidad horaria, documentación disponible, etc. Básicamente, establece la relación entre ambas partes: proveedor y cliente. Un SLA identifica y define las necesidades del cliente con respecto a la capacidad del proveedor. Los SLA son un punto de referencia de mejora continua debido a que el poder medir adecuadamente los niveles de cumplimiento de los WS, estos pueden mejorar y aumentar sus índices de calidad [Wikipedia, 2013].

Un SLA entre el proveedor de servicios y clientes, asegura a los clientes que pueden obtener el servicio que se paga y se obliga al proveedor del servicio a alcanzar sus promesas de servicio. Para los proveedores no cumplir un SLA puede provocar graves consecuencias económicas, por lo tanto, los proveedores están interesados en obtener un buen entendimiento de la relación entre lo que se promete en un SLA y lo que su infraestructura de TI puede entregar. En [Jin, 2002] se presenta un modelo SLA y una herramienta para la simulación, la cual es útil al proveedor de servicios en la etapa de diseño de SLA. Al diseñar un SLA, el proveedor debe de

conocer las capacidades de su infraestructura de TI de tal forma que se puedan aprovechar estas capacidades al máximo. En la composición de servicios, un cliente que desea agregar a un servicio a su flujo de trabajo, debe de evaluar la información SLI “Service Level Indicators” antes de seleccionarlo dentro de un conjunto de proveedores de servicio que ofrecen la misma funcionalidad. Los proveedores de WS deben de publicar los Objetivos de Nivel de Servicio (SLO) para aquellos clientes que revisen la calidad de servicio que se está ofertando.

La automatización y monitoreo de SLA significa la mínima intervención humana, aunque automatizar resulta difícil, pues se requiere que la especificación del SLA deber ser precisa e inequívoca, además de contar con un motor que gestione o cubra todo el proceso de seguimiento o vigilancia del SLA [Sahai, 2002], [Keller, 2003] y [Sahai, 2002].

4.3 Composición de WS

4.3.1 Composición de WS mediante flujos de composición

Según los autores [Menascé, 2002], [Menascé, 2004], [Zeng, 2003] y [Jaeger, 2005], la composición de Servicios Web puede llevarse a cabo de forma estática y dinámica, la forma estática se puede dar en la fase de diseño o desarrollo del Servicio Web; en la composición dinámica se consideran los atributos de calidad de Servicios Web para tal tarea, la composición dinámica puede llevarse a cabo mediante flujos de composición, por ejemplo, cíclico, paralelo, secuencial entre otros, cabe mencionar que para cada flujo de composición las métricas son diferentes. En el trabajo de [Menascé, 2002] se presentan tanto los sitios web y los sitios de Servicios Web. Los sitios web, tradicionalmente implementan todos los componentes necesarios para llevar a cabo transacciones de los usuarios: interface de usuario y administración de navegación, lógica del negocio y acceso a almacenamiento persistente. Los sitios de WS dan a los usuarios acceso a algunos o todos estos servicios a través de la Web.

La medición de los atributos de calidad es observada por usuarios de los WS. Estos usuarios no son seres humanos, sino programas que envían peticiones por servicios a proveedores de Servicios Web. Los problemas de QoS de WS han de ser evaluados tanto desde la perspectiva del proveedor como del cliente, del lado del proveedor, éste considera algunos aspectos de QoS, uno de ellos es la política de QoS. Esta política no es más que las condiciones de uso del WS que se traduce en SLA, desde el punto de vista del usuario, la calidad del WS es afectada debido al uso de otros WS, por lo tanto se debe de monitorear la calidad de servicio del Servicio Web que se está recibiendo. Es importante mencionar que un sitio que usa Servicios Web puede necesitar considerar Servicios Web transaccionales, es decir, requerir distribuir transacciones para tener las propiedades de *Atomicity*, *Consistency*, *Isolation* y *Durability* (ACID) en la presencia de algún tipo de sitio o fallas de red.

El autor [Menascé, 2004] argumenta que la exacta definición y proceso de medición para cada métrica debe ser bien definida para dar al consumidor del servicio y proveedores un común entendimiento, además de que en el proceso de medición para cada métrica se debe considerar: a) que medir (conjuntos o percentiles), b) como medir (frecuencia, intervalos y herramientas), c) quien hace la medición (el proveedor del servicio o agentes independientes externos), d) dónde se toman las medidas (punto de acceso del WS, red de conocimiento o en el consumidor). Así también en [Menascé, 2004], se presentan algunos escenarios de composición, por ejemplo, a) invocación del servicio probabilístico, b) invocación del servicio en paralelo, c) invocación del servicio uno antes de muchos mutuamente exclusivos predecesores terminados, d) ejecución del servicio después de que el primer predecesor se complete y e) ejecución del servicio después de que todos los predecesores completos. Así también [Menascé, 2004], presenta métricas para tiempo de ejecución y el costo total de la composición para los escenarios de: invocación probabilística, invocación paralela, activación sincronizada.

En [Zeng, 2003] se emplean diagramas de estados para especificar el modelo de procesos de un servicio compuesto y para la selección de servicios se emplearon métodos de programación lineal eficiente, se mencionan métricas tanto para selección y composición. Para la composición de servicios en el trabajo de [Zeng 2003] se emplea un mecanismo que determina la calidad de servicio de una composición mediante la agregación de atributos de calidad de servicios individuales. Con agregación de atributos de calidad se puede verificar si un conjunto de servicios satisfacen los requerimientos para toda la composición o no. El autor [Jaeger, 2005] argumenta que la agregación está basada en patrones de composición abstracta, que básicamente modela elementos estructurales de una composición como son rutas en paralelo, secuencia o ciclos. La calidad de un servicio compuesto se puede determinar mediante la agregación de QoS de los servicios individuales que lo componen, calculada por funciones de agregación de QoS.

4.3.2 Composición de WS usando Algoritmos

En la composición de servicios se hace uso de algoritmos genéticos como se menciona en [Canfora, 2005] y [Wang, 2007], cabe destacar que el autor Wang hace uso del algoritmo de recocido simulado, aunque no se presenta ningún resultado de comparación entre estos dos algoritmos. Así mismo se presenta un modelo de QoS el cual comprende los siguientes atributos de calidad: precio, tiempo de respuesta, capacidad máxima y reputación. En [Canfora, 2005], los algoritmos genéticos son comparados con los de la programación lineal entera, que es el enfoque más ampliamente adaptado. Este enfoque basado en algoritmos genéticos tiene como objetivo el determinar rápidamente un conjunto de servicios concretos a obligarse a los servicios abstractos componiendo el flujo de trabajo de un servicio compuesto. El uso de algoritmos en la composición de servicios tiene como objetivo facilitar y optimizar el proceso de composición, el más comúnmente adaptado es el enfoque de programación lineal entera.

4.3.3 Composición de WS mediante agentes o middleware

Dentro de la selección y composición dinámica de WS se encuentran los agentes o middleware los cuales facilitan estas tareas. En [Yu, 2005] se propone un marco basado en agente para facilitar la integración dinámica y adaptación de Servicios Web compatibles con QoS con restricciones de QoS punto a punto. Las principales funciones de un agente dinámico incluyen la adaptación, selección, composición y colección de servicio. El estudio que Yu realiza considera características funcionales y características QoS de Servicios Web para identificar las óptimas soluciones de procesos de negocios. Por otra parte en [Zeng, 2004] se presenta una plataforma middleware que aborda el tema de la selección de Servicios Web con el fin de que su composición maximice la satisfacción del usuario expresada como funciones de utilidad sobre atributos de QoS, mientras satisface las restricciones establecidas por el usuario y por la estructura del servicio compuesto. Realizar la tarea de composición dinámica de WS es posible a través de agentes de servicio, estos reciben las restricciones tanto funcionales y no funcionales de los clientes. Dentro de la composición de servicios es usada la función de utilidad (véase el punto 4.7.3), esta maximiza la satisfacción del usuario.

4.3.4 Composición de servicios bajo otros dominios

Para describir los requerimientos de QoS de clientes, en [Chen, 2006] se propone un nuevo lenguaje EWSDL (Enhance Web Service Description Language) de descripción del Servicio Web y también se propone optimizar el marco de Servicio Web concurrente con un evolucionado rol de proveedor para satisfacer requerimientos no funcionales de clientes, con el propósito de realizar composiciones. Así como también, Chen incluyó el uso del algoritmo CPA (Critical Path Algorithm) para la realización de la composición de servicios. La importancia de incluir los atributos de calidad en el servicio de descubrimiento de WS es para facilitar la tarea a mecanismos de descubrimiento basados en QoS, sin embargo estos todavía fallan en considerar dos puntos distintos. Primeramente, más soluciones involucran semánticas pobres así como diferentes terceras personas pueden tener diferente entendimiento de algunas métricas de QoS. Segundo existen numerosas métricas de QoS de WS, ambiguas e impredecibles que limitan la flexibilidad y expansibilidad. Para abordar estos dos problemas, en [Li, 2009] se presenta un enfoque novedoso y extensible basado en ontologías para la descripción de las limitantes o restricciones de QoS. Además se propone un método de cálculo de QoS para la composición de servicios para juzgar si el conjunto de servicios recomendados está calificado para las restricciones de QoS. El QoS de WS basado en ontología ofrece un enfoque y un algoritmo de predicción para la composición de WS. Este enfoque brinda un eficiente método de selección de servicios compatibles con QoS en el registro de servicios.

4.4 Reutilización de WS

Para la reutilización de servicios se emplea o se hace uso del término “componente web” que empaqueta servicios elementales en conjunto o complejos y presenta sus interfaces y operaciones en una consistente y uniforme manera en forma de una definición de clases. Los componentes web pueden ser reusados, especializados o extender Servicios Web complejos o elementales. En [Yang, 2002], se emplea el concepto de “componente web” y en [Yang, 2003] se emplea como “componente de servicio” pero se refieren al mismo concepto.

El desarrollo basado en el enfoque SOA, puede ser construido como servicios los cuales pueden ya existir dentro de una organización y pueden ser reusados cuando se construye una aplicación. La capacidad de reutilizar los servicios internos y externos dentro de un enfoque SOA, combinar servicios de formas nuevas y diferentes y añadir servicios adicionales donde sea necesario reduce el tiempo para desarrollar nuevas aplicaciones y llevarlos al mercado. [O'Brien, 2005]. El enfoque de arquitectura orientada a servicios (SOA) intenta fomentar la reutilización de componentes de software [Leymann, 2002].

La reutilización se da cuando en la composición de flujos de Servicios Web, estos pueden ser almacenados en la UDDI como nuevo servicio para que en el futuro, este pueda ser parte de otro servicio cuando así se requiera. Por otro lado, cuando las limitaciones de calidad y las preferencias son asignados a los servicios compuestos en lugar de los componentes de servicios, es fácil volver a utilizar el servicio compuesto, en los enfoques de “Quality Vector” y “Agent-oriented” [Frankova, 2005].

Desarrollar servicios bajo el enfoque SOA trae grandes beneficios, como por ejemplo, la reutilización de servicios existentes, el reutilizar servicios disminuye el tiempo de desarrollo de nuevas aplicaciones con mejores y nuevas características. Los servicios compuestos también pueden ser reutilizados dentro de otro servicio compuesto, pues la calidad de un servicio compuesto puede ser calculada mediante funciones de agregación.

4.5 QoS de WS en la nube computacional

4.5.1 Acerca de GRID y Cloud Computing

Grid Computing surge a mediados de los 90's para resolver problemas de cómputo a gran escala usando una red de máquinas que comparten recursos, como por ejemplo: supercomputadoras y grandes “clusters” dedicados en ese momento. La creación de Grid Computing es similar en forma y utilidad a la red de energía eléctrica. Algunos ejemplos de Grid Computing son: (TeraGrid, Open

Science Grid, caBIG, EGEE, Earth System Grid). Algunas organizaciones que definen estándares relevantes para Grid Computing: OGF, OASIS [Foster, 2008].

John McCarthy en los 60's previó que las instalaciones de computación serian proporcionadas al público en general como una herramienta o utilidad. La existencia de diferentes percepciones de Cloud Computing es debido a que, no es una tecnología nueva y que usa un conjunto de tecnologías existentes para ejecutar negocios en diferente forma, de hecho la virtualización y precio basado en utilidad no es nuevo [Zhang, 2010].

La popularidad de diferentes paradigmas varía con el tiempo. La popularidad de búsqueda en la web es medida a través de “Google search trends” [Buyya, 2008]. Algunos puntos de tendencia que describe el autor las menciona como sigue:

- A. IBM introduce “Blue Cloud” – Nov 15 2007.
- B. IBM, E.U. lanza “RESERVOIR” iniciativa de investigación para Cloud Computing, IT News Online –Feb 7 2008.
- C. Google y Salesforce.com en acuerdo de computación en nube, Siliconre-public.com –Abril 14 2008.
- D. Desmitificando el Cloud Computing, Intelligent Enterprise — Jun 11 2008.
- E. Yahoo se reorganiza para apoyar computación en nube ‘core strategies’, San Antonio Business Journal —Jun 27 2008.
- F. Yahoo, Intel y HP forman Cloud Computing labs, Reseller News —Jul 29 2008.

Otras tendencias similares a la anterior son presentadas en los trabajos de [Hashemi, 2012] y [Wang, 2010].

Cloud Computing ofrece servicios tanto para usuarios comunes y desarrolladores. Un ejemplo de uso común puede ser: la creación de una hoja de cálculo con el servicio de “Google Docs”. El cambio de los programas instalados localmente hacia Cloud Computing, afectará a todos los niveles del ecosistema de cómputo, desde el usuario casual a desarrollador de software, administrador de TI e incluso a fabricantes de hardware.

Cloud Computing tiene como objetivo concentrar la computación y el almacenamiento de datos, donde las máquinas de alto rendimiento están unidas por conexiones de gran ancho de

banda y todos estos recursos son manejados cuidadosamente. [Hayes, 2008] argumenta que el futuro de la Cloud Computing estará en los siguientes elementos:

- a) La palabra de moda para la Web: los tipos de aplicaciones de productividad que primero atrajo a la gente a las computadoras personales hace aproximadamente 30 años están apareciendo como servicios de software.
- b) Cómputo empresarial en la nube: Software para aplicaciones empresariales importantes (tales como atención al cliente, ventas y marketing) han sido generalmente ejecutadas en servidores de la empresa, pero varias compañías ahora ofrecen esto como un servicio bajo demanda, el primero fue Salesforce.com, fundada en 1999.
- c) Infraestructura de nube: está muy bien para externalizar la tarea de construir y mantener un centro de datos. Amazon.com se ha movido en este ecosistema de internet, “*Amazon Web Services*” ofrece almacenamiento de datos.
- d) Sistema operativo (SO) de nube: [Hayes, 2008] menciona un enfoque que proporciona todas las facilidades de un sistema operativo dentro de un navegador. Además, menciona que conforme la nube crece, es necesario que se adapte el movimiento de código abierto al nuevo modelo de computación. Una cosa es crear y distribuir un procesador de texto de código abierto y competir con Microsoft Word, y otra cosa es crear un Servicio Web para competir con Google Docs. En este sentido se deben tomar en cuenta cuestiones de seguridad, privacidad y confiabilidad.

En [Patidar, 2012] y [Zhang, 2010] existen diferentes conceptos para estructura o arquitectura de nube, pero en los trabajos de investigación más recientes se ha notado el uso de la definición dada por NIST [Mell, 2011]. Mientras [Zhang, 2010] divide en dos partes el tradicional rol del proveedor de servicio en un entorno computacional de nube: el proveedor de infraestructura quien administra las plataformas de nube y renta recursos de acuerdo a un modelo de precio basado en uso y el proveedor de servicio, quien renta recursos de uno o muchos proveedores de infraestructura para servir a usuarios finales.

En su artículo, [Zhang, 2010] también menciona que las empresas tratan de reorganizar sus modelos de negocio para obtener beneficios de este paradigma. Algunos de esos beneficios se pueden considerar al no realizar una inversión inicial en infraestructura, tener un bajo costo de operación y contar con servicios altamente escalables. La arquitectura de Cloud Computing que [Zhang, 2010] propone, está compuesta por cuatro capas: el hardware/datacenter, infraestructura, plataforma. Dentro de estas categorías la arquitectura de Cloud se distribuye de la siguiente forma: IaaS (Infrastructure as service), PaaS (platform as service) y SaaS (software as service). Estas

categorías también se mencionan en [Thakur, 2012] y [Patidar 2012], sin embargo este último no proporciona una descripción suficientemente genérica de una estructura de nube y sus componentes. De acuerdo a esta arquitectura, algunos atributos de calidad que se pueden considerar son disponibilidad, seguridad, desempeño, confidencialidad de datos y responsabilidad, nivel de bloqueo de datos, integración con IT interna. Mismos para los que debería existir forma de medirlos. En los aspectos de calidad, confiabilidad y desempeño, el usuario no tiene mayor alternativa actualmente que confiar en la promesa del proveedor. De acuerdo a [Patidar 2012] y a [Zhang 2010], estos aspectos representan un reto a solucionar dentro de la nube computacional.

A continuación se presenta en la tabla 3 una comparativa entre Grid y Cloud Computing, la comparación se realiza mediante sus principales características de ambos paradigmas.

Tabla 3: Grid Computing vs Cloud Computing [Hashemi, 2012].

Parametro	Grid Computing	Cloud Computing
Objetivo	Intercambio colaborativo de recursos.	Uso del servicio (elimina los detalles).
Enfoque computacional	Operaciones computacionalmente intensivas.	Instancias de alto nivel y estándar.
Gestión de flujo de trabajo	En un nodo físico.	Instancia en EC2 (Amazon EC2+S3).
Nivel de abstracción	Bajo (más detalles).	Alto (elimina detalles).
Grado de escalabilidad	Normal.	Alto.
Multitarea	Sí.	Sí.
Transparencia	Bajo.	Alto.
Tiempo de ejecución	No en tiempo real.	Servicios en tiempo real.
Tipo de petición	Pocos pero asignación grande.	Lotes de asignación pequeña.
Unidad de asignación	Trabajo o tarea (pequeño)	Todas las formas y tamaños (ancho y estrechos).
Virtualización	No es un producto.	Vital.
Portal accesible	A través de un sistema de DNS.	Únicamente usando IP (sin DNS registrado).
Transmisión	Sufre de retrasos de internet.	Fue significativamente rápido.
Seguridad	Baja (servicio de red certificado).	Alta (virtualización).
Infraestructura	Comando de bajo nivel.	Alto nivel de servicios (SaaS).
Sistema Operativo	Algún SO estándar	Un "hypervisor" (VM) sobre el que se ejecutan multiples SOs.
Propiedad	Múltiple.	Individual.
La interconexión de la red	Principalmente Internet con latencia y bajo ancho de banda.	Dedicado, de gama alta con baja latencia y gran ancho de banda.
Descubrimiento	Indexación centralizada y servicios de información descentralizada.	Servicios de Membresía.
Negociación de servicio	Basado en SLA.	Basado en SLA.

Administración de usuario	Descentralizado y también basado en organización virtual (VO por sus siglas en inglés).	Centralizado o se puede delegar a terceros.
Administración de recurso	Distribuido.	Centralizado/Distribuido.
Asignación/Programación	Descentralizada.	Ambas Centralizada/Descentralizada.
Interoperabilidad	Estándares abiertos del foro de Grid.	Servicios Web (SOAP and REST).
Administración de fallas	Limitado (a menudo fallas de tareas/aplicaciones son reiniciadas).	Fuerte (VMs pueden ser fácilmente migradas desde un nodo a otro).
Precios de los servicios	Dominada por el bien público o en privado asignado.	Fijación de precios de utilidades, descuentos para grandes clientes.
Fácil de usar	Bajo.	Alta.
Tipo de servicio	CPU, red, memoria, banda ancha, dispositivos, almacenamiento,...	IaaS, PaaS, SaaS, cualquier cosa como un servicio.
Almacenamiento intensivo de datos	Adecuado para eso.	No adecuado para eso.
Ejemplo del mundo real	SETI, BOINC, Folding@home, GIMPS.	Amazon Web Service (AWS), Google apps.
Tiempo de respuesta	No puede proporcionar servicio a la vez y necesitan ser programados.	En tiempo real.
Objeto crítico	Recurso computacional.	Servicio.
Número de usuarios	Pocos.	Muchos.
Recurso	Limitado (debido a que el hardware es limitado).	Ilimitado.
Configuración	Difícil, ya que los usuarios no tienen privilegios de administrador.	Muy fácil para configurar.
Futuro.	Cloud Computing.	La siguiente generación de internet.

Otras comparaciones similares se presentan en los siguientes trabajos: [Wang, 2010], [Vaquero, 2009] y [Foster, 2008].

De acuerdo a la tabla 3 se puede observar que la principal característica de la nube computacional es que se brindan recursos virtualizados como servicios, mientras que en la Grid Computing no sucede lo mismo.

4.5.2 Acerca de seguridad en Cloud Computing

En esta tesis se adentra un poco sobre la seguridad o privacidad de los datos en la nube, ya que, se considera un punto importante a tratar y dar al lector una idea sobre seguridad lo que implica migrar hacia la nube computacional, diferentes puntos de vista con respecto a seguridad se

discuten en [Harauz, 2009],[Kaufman, 2010],[Mendhe, 2012],[Pearson, 2009],[Pearson, 2011] y [Yao, 2010].

Desde una perspectiva técnica, Cloud Computing incluye SOA y aplicaciones virtuales tanto para hardware y software. Debido a que en la Cloud Computing se comparten máquinas virtuales para desempeñar actividades diarias de diferentes empresas, este entorno requiere un nivel implícito de confianza así como un nivel explícito de vigilancia para asegurar el éxito. Para asegurar la Confidencialidad, Integridad y Disponibilidad (CID) de datos, el proveedor de almacenamiento debe proporcionar capacidades como mínimo, incluir: un esquema de encriptación probado para asegurar que el entorno de almacenamiento compartido salvaguarda todos los datos, controles de acceso estrictos para prevenir accesos no autorizados para los datos y copia de seguridad de datos programadas y almacenamiento seguro de los medios de “backup”. Para abordar los problemas de seguridad, el autor menciona que, se debe desarrollar un modelo de seguridad que promueva la Confidencialidad, Integridad y Disponibilidad, que permita a cada nube ofrecer una medida de su actual y proyectada conocida como CID.

Es importante también mencionar que el nivel de seguridad dependerá del grado de confidencialidad de los datos o información. Cloud Computing también debe preocuparse por la seguridad cibernética, si un cibercriminal puede identificar que proveedor es el más vulnerable, este proveedor se convierte en un objeto altamente visible, por lo tanto esta falta de seguridad asociada con esta sola entidad o proveedor amenaza a toda la nube en la que reside. Para asegurar la seguridad, según el autor [Harauz 2009], se deben de tomar medidas proactivas, por ejemplo, adoptar estándares universales para asegurar la interoperabilidad entre proveedores de servicios, así como también el desarrollo de estándares de seguridad para asegurar la CID de datos.

La preocupación por privacidad continuará creciendo debido a que bases de datos a menudo contienen información personal y sensitiva relacionada a empresas y/o particulares. Es por eso que hay una creciente conciencia de la necesidad de diseño para la privacidad de las empresas y organizaciones no gubernamentales. Para ello la infraestructura de la nube computacional ofrece una gama de servicios de privacidad con garantías en cuanto al grado de privacidad que ofrece, por ejemplo, mecanismos para la garantía de privacidad en el lado del proveedor del servicio y suministro de sello de privacidad de WS, aunque no se describen estos dos servicios de seguridad [Pearson 2009].

Con respecto a seguridad en la virtualización de los recursos que ofrece la nube es también relevante. Un estudio de Prism Microsystems, en el cual, más de la mitad de los encuestados creen que la vitalización creará una nueva capa que podría ser atacada y que la proliferación de entornos virtualizados reduciría la visibilidad de la seguridad. Tradicionalmente, los centros de datos consisten en una gran colección de conjuntos de servidores implementando

medidas de seguridad de perímetro incluyendo firewalls, zonas desmilitarizadas², sistemas de prevención y detección de intrusión y herramientas de monitoreo de red; mientras que en la virtualización el concepto de perímetro de red desaparece, debido a que el entorno virtual soporta múltiples máquinas virtuales de varias organizaciones sobre un solo servidor se debe proporcionar un nivel de seguridad a la VM en vez del perímetro [Kaufman, 2010].

El paradigma de la virtualización, en sí mismo introduce riesgos de seguridad debido a que el acceso remoto proporciona la exposición a potenciales “cyberattackers”. La naturaleza dinámica de las VMs permite la fácil reconfiguración, lo cual, propaga vulnerabilidades y errores de configuración desconocidos y mantener registros del estado de seguridad de toda la nube en algún momento dado es difícil, si no imposible. Otra de las vulnerabilidades únicas es que cuando una VM esta offline, está aún disponible para alguna aplicación que puede acceder al servidor físico sobre el que este reside. Un usuario remoto de una VM puede acceder a otra VM inactiva si ambas residen sobre el mismo servidor físico. Además de que cuando una VM está inactiva, esta no puede escanear por malware y por lo tanto son altamente susceptibles, esta vulnerabilidad no está restringida a las VM en un hipervisor particular, es por eso que se necesita mover la seguridad del perímetro a la VM.

El entorno Cloud Computing continuamente es amenazado por códigos maliciosos que pueden interferir con el “hypervisor” u otras máquinas virtuales, en [Mendhe, 2012] se estudian técnicas matemáticas usadas en la literatura para modelar la propagación de códigos maliciosos, principalmente gusanos. El autor presenta una lista de requerimientos de seguridad para Cloud Computing, así como también una lista de los más importantes ataques que pueden ser conducidos contra la infraestructura de nube y por último se presentan algunas técnicas de seguridad de nube, todo esto como medidas de respuesta a los ataques a la nube.

La “*Accountability*” (responsabilidad o rendición de cuentas) es un punto clave para abordar problemas de confianza y complejidad en la nube computacional. La confianza y complejidad están estrechamente ligadas: un proveedor de servicio en la nube tiene responsabilidades legales y éticas para asegurar la privacidad y protección de datos, consiguiendo así un carácter fidedigno de sus servicios. “*Accountability*” es útil para proteger información confidencial y sensible, incrementando la confianza del consumidor, clarificando la situación legal en la nube computacional y facilitar las transferencias transfronterizas de datos. Con respecto a normas de ley, es probable que con el tiempo, la legislación ponga más énfasis en la rendición de cuentas o responsabilidad de servicio [Pearson, 2011].

Un mecanismo o servicio para lograr “*Accountability*” de los Servicios Web dentro de la nube computacional dentro del proceso de composición es el sistema llamado

² [http://es.wikipedia.org/wiki/Zona_desmilitarizada_\(inform%C3%A1tica\)](http://es.wikipedia.org/wiki/Zona_desmilitarizada_(inform%C3%A1tica))

“*TrustworthysErviceful Cloud*” (TEC), con tal sistema se puede hallar la raíz de una violación, falla o mal funcionamiento de un nodo. La raíz puede siempre ser identificada y ser asociada con la entidad responsable, esta asociación es soportada por una evidencia no impugnada [Yao, 2010].

La preocupación por la seguridad y privacidad de los datos dentro de la nube computacional es un tema importante tanto para los usuarios comunes y empresas que confían sus datos en este tipo de infraestructura, cabe mencionar que el grado de seguridad de los datos, dependerán del grado de confidencialidad de estos. Seguridad y privacidad en la Cloud Computing necesitan ser tratados de manera diferente. Los temas más importantes sobre seguridad en la nube son: infraestructuras de nube, análisis de amenazas, riesgos de infraestructuras y aislamiento de datos y software [Mendhe, 2012].

4.5.3 Selección de Servicios Web en Cloud Computing

Con la selección de Servicios Web en la nube se persiguen los enfoques de selección de servicios comúnmente usados en la arquitectura SOA, cabe aclarar que a diferencia de los servicios desarrollados en la tradicional SOA, en Cloud Computing, no se cuenta con una UDDI como repositorio de los servicios según los autores [Jensen, 2009] y [Wei, 2010] debido a que no se ha desplegado o adoptado ampliamente este estándar en la nube.

Por otro lado, autores [Chen, 2010] y [Falasi, 2011] proponen diferentes enfoques que puedan solucionar el problema de descubrimiento de WS en la nube. En [Chen 2010] se propone un modelo de registro de servicio llamado SRC “*Service Registry on Cloud*” para proporcionar servicio de descubrimiento compatibles con atributos de calidad y compatibles con comportamiento. La diversidad de servicios en la nube brinda el reto de: ¿Cómo descubrir el servicio más adecuado? El cual, en las peticiones de servicio de descubrimiento de servicios incluye requerimientos funcionales y no funcionales. Para los requerimientos funcionales, el enfoque de mejora del servicio de descubrimiento involucra agregar semántica a las descripciones de Servicios Web y entonces registrar estas descripciones en los registros.

Para los requerimientos no funcionales, la información sobre la calidad de servicio es la base para procesar las peticiones de descubrimiento de servicio. Hay dos tipos de información de QoS: valoración estática de QoS y estado dinámico de QoS. Para soportar tanto descubrimiento de servicio compatible con QoS y compatible con comportamiento, el servicio de registro debería de tener suficiente espacio de almacenamiento para almacenar los datos necesarios, además de ser una aplicación de uso intensivo de datos. [Falasi, 2011] presenta un marco de trabajo de composición de servicios en la nube para enfrentar todos los aspectos del proceso de composición de servicios empezando desde la negociación dinámica de SLAs, además de la verificación de servicios en tiempo real y en una composición. El marco está compuesto por las siguientes partes:

un directorio Cloud de terceros, proveedores Cloud y un agente compositor de confianza. Debido a que en la nube computacional no existe un repositorio UDDI según los autores [Jensen 2009], [Wei 2010] y [Goscinski, 2010], en su lugar se cuenta con un directorio Cloud de terceros.

Con respecto a las métricas de selección de servicios elementales, es predominante los QoS de “Price”, “Response time”, “Throughput”, “Reputation”, “Availability”, “Reliability” que pueden ser medidos con las métricas que se presenta en el punto 4.7.2 de esta tesis.

4.5.4 Composición de WS en Cloud Computing

Los autores [Klein, 2012],[Cui, 2012],[Wang, 2011],[Alrifai, 2009],[Ardagna, 2007] y [Yu, 2007], mencionan que en la composición de WS comúnmente usan diferentes algoritmos que permiten o facilitan la composición, algunos están enfocados en optimizar el tiempo de cálculo en la composición o reducir el espacio de búsqueda. El algoritmo más común usado en la solución de problemas de optimización de composición es la programación entera mixta para identificar el servicio web más conveniente según los autores [Alrifai, 2009] y [Wang 2011] , así como también se menciona el uso de algoritmos genéticos, viterby, MMKP para la selección de servicios, es importante mencionar que, en la composición de servicios existen flujos de composición, como lo son: secuencial, ciclo, condicional, paralelo; siendo el secuencial el más común.

Diferentes enfoques pueden ser usados para la composición de servicios, tal como el método llamado LOEM compatible con QoS para mejorar la eficiencia de composición de servicios cuando se dispone de numerosas soluciones compuestas [Qi, 2011]. Cuando se especifican los requerimientos no funcionales es posible que se haga de forma estricta o no, para tal caso, [Rosenberg, 2009] presenta un enfoque diferente a aquellos que sólo consideran limitantes estrictas durante la optimización, pues el enfoque presentado en este trabajo permite usar tanto limitantes estrictas como flexibles para especificar QoS. El primero indica las limitaciones necesarias que deben cumplirse durante el proceso de composición mientras que el segundo representa restricciones opcionales de QoS que son "bueno tener" y que deben ser satisfechas si es posible. En este enfoque propuesto no optimiza la composición para encontrar la mejor solución global, sin embargo, se trata de encontrar una solución óptima dentro de los límites de restricción dados por el usuario como parte de la especificación de servicio compuesto.

El trabajo de [Rosenberg, 2009], se centra en las composiciones compatibles con QoS a nivel microflujo, es decir, de ejecución corta de servicios compuestos. Se usa un lenguaje de especificación de QoS llamado “VCL” (lenguaje de composición Viena) en composiciones de servicios, así como, el uso de técnicas y algoritmos para transformar y optimizar una composición especificada en VCL. El enfoque se evaluó en términos de su rendimiento para demostrar que la

optimización es factible para formar la base para la composición en tiempo de diseño y recomposición en tiempo de ejecución.

En [Alrifai, 2010] se presenta un enfoque de selección de servicio llamado “skyline” el cual permite identificar aquellos servicios que no están denominados por algún otro servicio funcionalmente equivalente y que por lo tanto son candidatos válidos para la composición. Se presenta un método para determinar que niveles de QoS de un servicio deberían ser mejorados de manera que no esté denominado por otros servicios.

4.6 Otra información relevante a trabajos sobre QoS de WS

En esta sección, se presenta información relevante a trabajos sobre QoS de Servicios Web, tal como: la actividad del grupo de trabajo de la W3C y un modelo de calidad de WS propuesto por OASIS.

4.6.1 Actividad del grupo de trabajo de la W3C

4.6.1.1 ¿Qué es la W3c?

El consorcio World Wide Web (W3C) es una comunidad internacional donde organizaciones miembros, personal a tiempo completo y el público en general trabajan conjuntamente para desarrollar estándares Web. Liderado por el inventor de la Web Tim Berners-Lee y el Director Ejecutivo (CEO) Jeffrey Jaffe, la misión del W3C es guiar la Web hacia su máximo potencial [W3C, 2012d].

4.6.1.2 Acerca de los grupos de trabajo de la W3C

Los grupos de trabajo se caracterizan por producir los resultados por ejemplo, las normas de seguimiento de los informes técnicos, software, series de pruebas y revisiones de los entregables de otros grupos [W3C, 2012e].

En esta tesis nos enfocaremos a dos grupos de trabajo de la W3C con respecto a Servicios Web:

- Política de Servicios Web (en inglés, Web Services Policy) [W3C, 2012a].
- Acceso a los recursos de Servicios Web (en inglés, Web Services Resource Access) [W3C, 2012b].

4.6.1.3 Estado de actividad de los Servicios Web

Los Servicios Web proporcionan un medio estándar de interoperabilidad entre diferentes aplicaciones de software, que se ejecuta en una variedad de plataformas y/o marcos. Los Servicios Web se caracterizan por su gran interoperabilidad y extensibilidad, así como ser procesables por máquina, gracias a las descripciones del uso de XML. Se pueden combinar en una forma imprecisa con el fin de lograr operaciones complejas. Programas que prestan servicios simples pueden interactuar unos con otros con el fin de ofrecer sofisticados servicios de valor añadido [W3C, 2012c].

La actividad de la W3C en Servicios Web es el diseño de la infraestructura, la definición de la arquitectura y la creación de las tecnologías principales para los Servicios Web. El protocolo SOAP 1.2 basado en XML, marco de mensajería que se convirtió en una recomendación del W3C en junio de 2003 y el mecanismo de optimización de transmisión de mensaje SOAP en enero de 2005.

Dentro de las tecnologías creadas, podemos encontrar las siguientes más importantes:

- SOAP Version 1.2 Part 0: Primer (Second Edition) (W3C Recommendation 27 April 2007)
- Web Services Description Language (WSDL) Version 2.0 Part 1: Core Language (W3C Recommendation 26 June 2007)
- Web Services Addressing 1.0 – Core (W3C Recommendation 9 May 2006)
- Web Services Policy 1.5 – Framework (W3C Recommendation 04 September 2007)
- Web Services Event Descriptions (WS-EventDescriptions) (W3C Recommendation 13 December 2011)
- Web Services Enumeration (WS-Enumeration) (W3C Recommendation 13 December 2011)
- SOAP over Java Message Service 1.0 (W3C Recommendation 16 February 2012)

4.6.2 Modelo de calidad de WS

4.6.2.1 WSQM

Los modelos de calidad de software se mencionan en el anexo A.1 están enfocados a la calidad del producto de software, a diferencia de estos, WSQM está enfocado a la calidad de servicio de WS. A continuación se hace mención del modelo de calidad de Servicios Web propuesto por OASIS, llamado WSQM (Web Service Quality Model), el cual está dividido en tres partes como se ve en la figura 3.

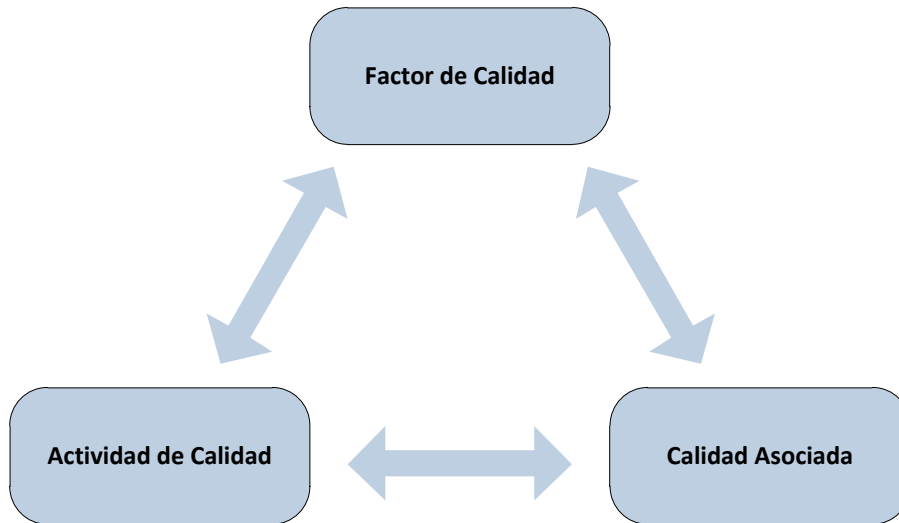


Fig. 3: Modelo de Calidad de Servicios Web, WSQM de OASIS [OASIS, 2005].

El modelo de calidad de Servicios Web de la figura 3 es llamado WSQM, el cual está basado en el estándar ISO9126 que se menciona en el anexo A.1, desarrollado por OASIS (*“Organization for the Advancement of Structured Information Standards”*) y está dividido en tres partes como se aprecia en la figura 3.

Factor de Calidad: es un componente fundamental que reconoce la calidad de Servicios Web para gestionar esta calidad. Es decir, describe las características y sub-características y atributos de calidad de Servicios Web.

Actividad de Calidad: se refiere a los distintos modelos de acción realizada por la Calidad Asociada para la estabilidad de la calidad de los Servicios Web.

Calidad Asociada: se refiere a los roles o tareas de las organizaciones o personas relacionadas a los Servicios Web, en este caso relacionadas a la actividad de calidad.

Los Servicios Web son considerados como un producto usado remotamente, así que la calidad de Servicios Web debería ser revisada cuando ellos son usados en sitios remotos.

4.6.2.2 Calidad de Servicios Web como un Servicio

En [OASIS, 2010], se menciona que los Servicios Web son proporcionados como un servicio, no como un producto en sí. Considerando las características orientadas a los servicios de Servicios Web, el Modelo de Calidad de Servicios Web debe establecerse desde el punto de vista de calidad del servicio, no del producto.

La calidad de Servicios Web es literalmente la calidad de uso de Servicios Web y dependiendo de los puntos de vista de uso de un servicio, este puede ser considerado en tres capas:

- Capa Nivel de Negocios (Business Level Layer).
- Capa Nivel de Servicio (Service Level Layer).
- Capa Nivel de Sistema (System Level Layer).

Cada capa tiene uno o varios sub-factores de calidad como se puede apreciar en la figura 4.

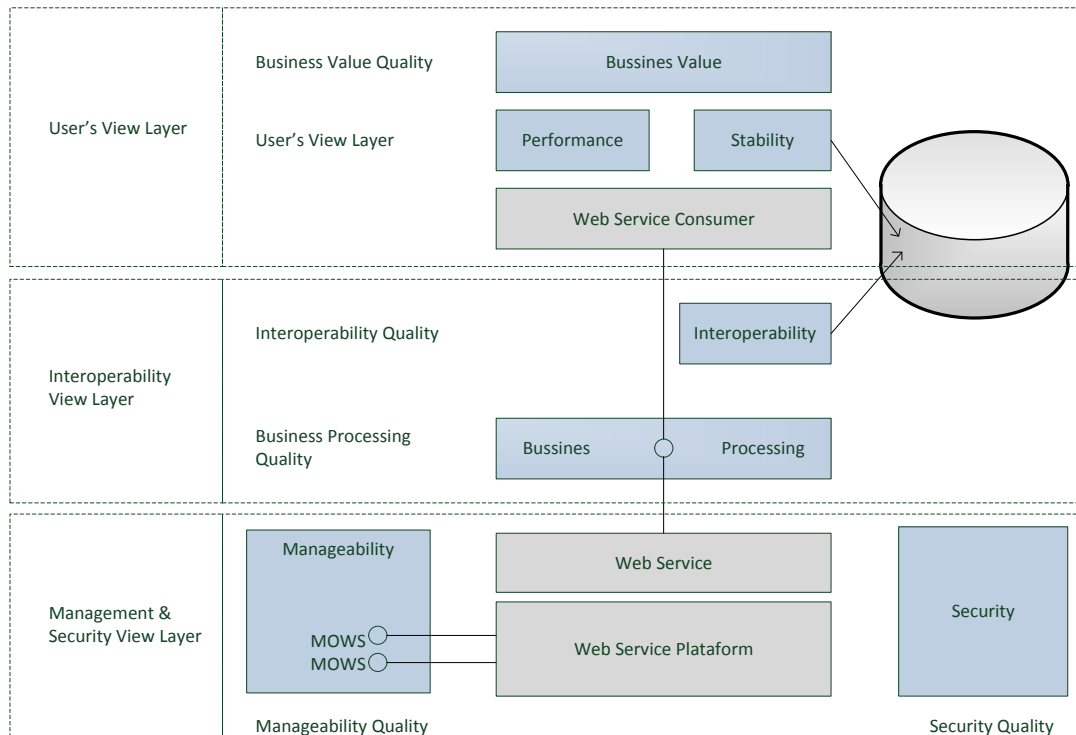


Fig. 4: Factores de Calidad de Servicios Web [OASIS, 2005].

La primera capa, Capa Nivel de Negocios es la calidad para representar el valor del negocio percibido por el usuario mientras usa el Servicio Web y es llamado Calidad del Valor de Negocio. En este nivel se identifican los atributos de calidad "Performance" y "Stability".

La segunda capa, Capa Nivel de Servicio es la calidad de desempeño medible del Servicio Web percibida por el usuario mientras usa el Servicio Web y es llamado Calidad Medible del Nivel del Servicio. Ésta calidad incluye asuntos o cuestiones tales como estabilidad y escalabilidad así como tiempo de respuesta. La calidad en la Capa Nivel de Usuario puedes ser medida cuando un usuario experimenta mientras usa los Servicios Web.

La tercera capa es la Capa Nivel de Sistema. Ésta capa puede ser dividida dentro de la “capa de interoperabilidad” y “capa de seguridad y administración”. La capa de interoperabilidad es la que determina si el Servicio Web, que es desarrollado en diferentes entornos de sistema por diferentes desarrolladores, puede propiamente interoperar. La calidad en esta capa puede ser subdividida en dos tipos de calidad dependiendo del interés del usuario. Uno es la calidad del formato del mensaje si entre servicios es intercambiable, es decir, si se ajusta al estándar y/o especificado por las organizaciones estándar, denominado Calidad de Interoperabilidad.

Otro es la calidad de los mensajes de interoperación si se ejecutan correctamente la lógica del negocio, llamado Calidad del proceso de Negocio. Esto puede ser subdividido entre “Mensajería Confiable” para la entrega de forma estable en alguna situación inestable de red y los factores de calidad para correcta ejecución de “contextos de los negocios” deseados.

La calidad en la capa de vista de interoperabilidad se puede obtener mediante la evaluación de “información de registro de mensajes (en inglés message log)” que se guarda mediante la intercepción de un mensaje en el medio de intercambio entre los Servicios Web y “el procesamiento de mensajes de estado de registro de la información (en inglés message processing status log information)” que se guarda, mientras un mensaje está siendo procesado.

La capa de administración y seguridad es para indicar la calidad de la gestión y la calidad de la seguridad. La calidad de administración es la calidad para indicar la capacidad de administración dentro o fuera del sistema. La calidad de seguridad es la calidad para indicar el nivel de la neutralización de los Servicios Web para el acceso no autorizado o ataque desde el exterior. La capacidad de gestión y calidad de la capa de seguridad puede ser comprobada por las pruebas de capacidad de administración y las funciones de seguridad.

Las 3 capas anteriores se componen de factores de calidad de Servicios Web, como se ve en la figura 5.

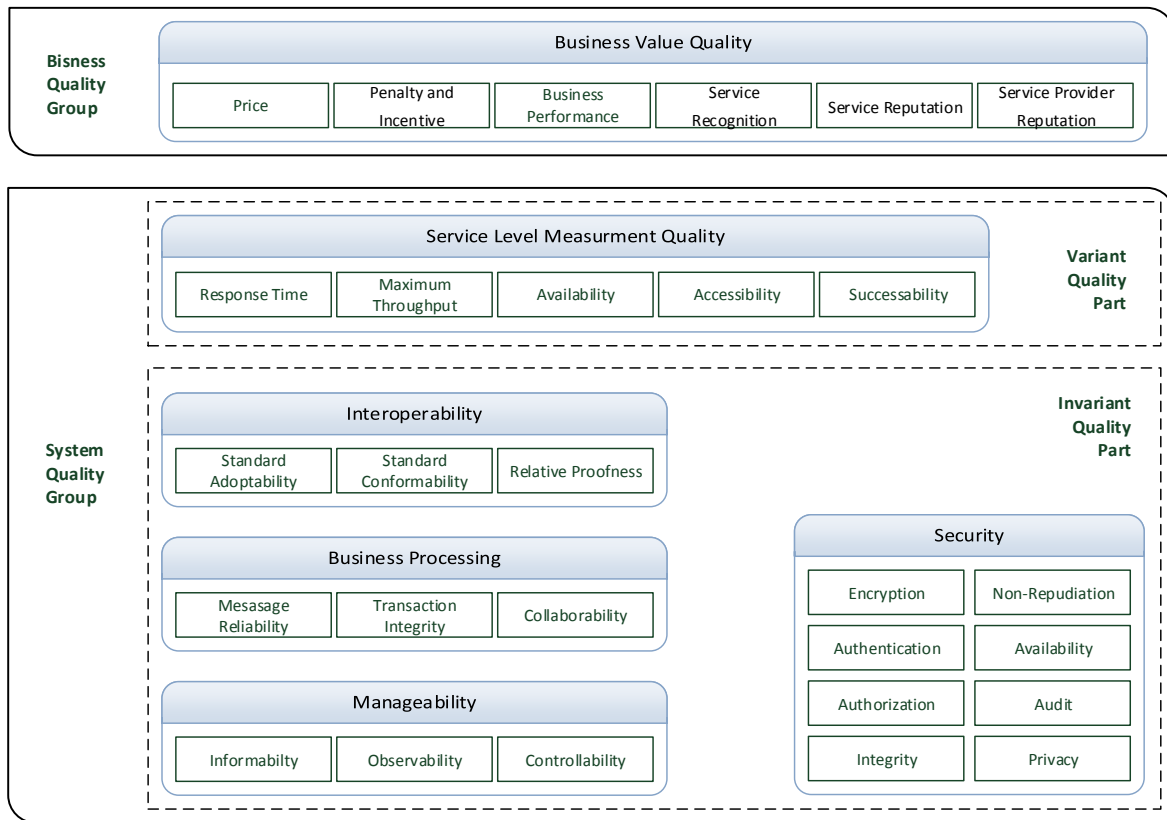


Fig. 5: Estructura de factor de calidad de WS.

Dentro de la capa “Service Level Measurement Quality” se presentan algunas métricas para la medición de estos factores, los cuales se describen en el punto 4.7.2.

Es importante también mencionar que en algunos trabajos de investigación sobre Servicios Web, cada autor propone su propio modelo de calidad de WS, es decir, agrupan a los atributos de calidad de forma diferente con sus métricas, así como también, proponen algunas métricas diferentes, como una contribución a este campo de investigación.

4.7 Atributos de calidad y Métricas

4.7.1 Conceptos o definiciones de atributos de calidad

A continuación, se presentan conceptos o definiciones de atributos de calidad de WS encontrados en la literatura, con la finalidad de observar o tener un criterio de cómo cada autor proporciona una definición para atributos de calidad.

Según [Nurhayati, 2008] se definen los siguientes atributos de calidad:

- **Service time.** Es la longitud de tiempo que toman los servicios para proporcionar una respuesta a varios tipos de peticiones.
- **Reliability.** Se refiere a la capacidad de mantenimiento del servicio y la calidad del servicio.
- **Execution Price.** Se refiere a la cantidad de dinero que el solicitante del servicio tiene que pagar por la ejecución de una operación.
- **Availability.** Se refiere a la presencia de un Servicio Web para un cliente para conectarse a él.
- **Performance.** Es medido por el rendimiento y latencia. El desempeño puede también ser determinado por el tiempo de respuesta para garantizar el tiempo máximo requerido para completar una solicitud de servicio.
- **Security.** Se refiere a mecanismos de autenticación, encriptación de mensajes y control de acceso, confidencialidad, no repudio y resistencia a los ataques de denegación del servicio.
- **Accessibility.** Se refiere a la capacidad de satisfacer la petición de un servicio.
 - **Transaction.** Se relaciona con propiedades ACID, que contienen las siguientes características.
 - **Atomicity.** Ejecuta las transacciones enteras o nada en absoluto.
 - **Consistency.** Mantiene la integridad y consistencia de los datos en la operación de actualización.
 - **Isolation.** Transacciones individuales corren como si otras transacciones no están presentes.
 - **Durability.** Es la persistencia de resultados.
- **Capacity.** Es el número máximo de peticiones concurrentes que el servidor puede procesar para garantizar el desempeño o el número de conexiones concurrentes que es permitido por el servicio.
- **Integrity.** Se refiere al mantenimiento de la correcta y consistente interacción de la fuente.

- **Regulatory.** se refiere a la conformidad y cumplimiento de las reglas, leyes, estándares y especificaciones.
- **Reputation.** Mide la integridad del servicio basado sobre la experiencia de uso del servicio de usuarios.

Según [W3C, 2003] se definen los siguientes atributos de calidad:

- **Performance:** El desempeño de un Servicio Web representa la rapidez que la petición a un servicio puede ser completada. Esto puede ser medido en términos de rendimiento, tiempo de respuesta, latencia, tiempo de ejecución y tiempo de transacción, etc.
- **Reliability:** Los Servicios Web deberían ser proporcionados con alta confiabilidad. La confiabilidad aquí representa la habilidad de un Servicio Web para desempeñar estas funciones requeridas bajo condiciones de estado para un intervalo de tiempo especificado. La confiabilidad es medida (métrica) de un Servicio web para mantener la calidad del servicio. La medida global de un Servicio Web es relacionada con el número de fallas por día, semana, mes, o año. Fiabilidad es también relacionado a la entrega segura y ordenada de mensajes que se transmiten y reciben por solicitantes de servicio y proveedores de servicio.
- **Scalability:** Los Servicios Web deberían ser proporcionados con alta escalabilidad. La escalabilidad representa la capacidad de incrementar la capacidad de cómputo del servicio del sistema de cómputo del proveedor y la habilidad del sistema para procesar más peticiones de usuario, operaciones o transacciones en un intervalo de tiempo dado. Esto también está relacionado para el desempeño. Los Servicios Web deberían ser escalables en términos de número de operaciones o transacciones soportadas.
- **Capacity:** Los Servicios Web deberían ser proporcionados con la capacidad requerida. Capacidad es el límite del número de peticiones simultáneas que deben contar con garantía de desempeño. Los Servicios Web deberían soportar el número requerido de conexiones simultáneas.
- **Robustness:** Los Servicios Web deberían ser proporcionados con alta robustez. Robustez aquí representa el grado para que un Servicio Web pueda funcionar correctamente aun en la presencia entradas invalidas, incompletas o en conflicto. Los Servicios Web deberían todavía trabajar aun si parámetros incompletos son proporcionados a la invocación de solicitud de servicio.

- **Exception handling:** Los Servicios Web deberían ser proporcionados con la funcionalidad de manejo de excepciones. Ya que esto no es posible para el diseñador del servicio especificar todos los posibles resultados y alternativas (especialmente con varios casos especiales y posibilidades imprevistas), excepciones deberían ser manejadas adecuadamente. El manejo de excepciones está relacionado a como el servicio maneja estas excepciones.
- **Accuracy:** Los Servicios Web deberían ser proporcionados con alta precisión. La precisión aquí es definida como la tasa de error generado por el Servicio Web. El número de errores que el servicio genera sobre un intervalo de tiempo debería ser minimizado.
- **Integrity:** La integridad para los Servicios Web debería ser proporcionada de modo que un sistema o componente puede prevenir accesos o modificaciones no autorizados a programas o datos de computadora. Puede haber dos tipos de integridad: integridad de datos y transaccional. La integridad de datos define si la transferencia de datos es modificada durante la transmisión. La integridad transaccional se refiere a un procedimiento o conjunto de procedimientos, que es garantía para preservar la integridad de la base de datos en una transacción.
- **Accessibility:** Los Servicios Web deberían ser proporcionados con alta accesibilidad. La accesibilidad aquí representa si el Servicio Web es capaz de servir las peticiones de los clientes. Alta accesibilidad puede ser logrado, por ejemplo: mediante la construcción de sistemas altamente escalables.
- **Availability:** Los Servicios Web deberían estar listos (es decir, disponibles) para consumo inmediato. Esta disponibilidad es la probabilidad de que el sistema esté en marcha y relacionado con la fiabilidad. El tiempo de reparación está asociado con la disponibilidad. El tiempo de reparación (TTR) representa el tiempo que toma este en reparar el Servicio Web. El servicio debería estar disponible inmediatamente cuando este es invocado.
- **Interoperability:** Los Servicios Web deberían ser interoperables entre los diferentes entornos de desarrollo usados para implementar servicios de modo que desarrolladores usan estos servicios no tienen que pensar acerca que lenguaje de programación o sistema operativo en que los servicios están alojados.
- **Security:** Los Servicios Web deberían ser proporcionados con la seguridad requerida. Con el incremento en el uso de Servicios Web que son entregados sobre el público en internet, hay una creciente preocupación acerca de la seguridad. El proveedor del Servicio Web

puede aplicar diferentes enfoques y niveles de proporcionar la política de seguridad dependiendo sobre el solicitante del servicio.

La seguridad para los Servicios Web significa proporcionar autenticación, autorización, confidencialidad, trazabilidad/auditabilidad, encriptación de datos y no repudio. Cada uno de estos aspectos son descritos como sigue:

- **Autenticación:** Usuarios (u otros servicios) quienes pueden acceder al servicio y datos deberán ser autenticados.
- **Autorización:** Usuarios (u otros servicios) deberán ser autorizados de modo que únicamente ellos puedan acceder a los servicios protegidos.
- **Confidencialidad:** Los datos deberían ser tratados apropiadamente de modo que únicamente usuarios autorizados (u otros servicios) puedan acceder o modificar los datos.
- **Responsabilidades:** El proveedor puede tener responsabilidad para sus servicios.
- **Trazabilidad y Auditabilidad:** Esto debería ser posible trazar el historial de un servicio cuando un servicio fue solicitado.
- **Encriptación de datos:** Los datos deberían ser encriptados.
- **No repudio:** Un usuario no puede negar solicitar un servicio o dato después del hecho. El proveedor del servicio necesita asegurar estos requerimientos de seguridad.

Según [OASIS, 2005] y [OASIS, 2010] se definen los siguientes atributos de calidad:

Los atributos de calidad y Métricas que se mencionan a continuación son del modelo de calidad de Servicios Web llamado WSQM desarrollado por OASIS.

Estos sub-factores de calidad están dentro de la capa de Medición de Calidad del Nivel del Servicio (en inglés Service Level Measurement Quality). Esta capa a su vez se divide en Performance y Stability como se muestra en la figura 7. Mediciones de “*performance*” incluye los sub-factores de “response time” y “maximum throughput”. Mediciones de “*Stability*” incluye los sub-factores de “*Availability*”, “*Successability*” y “*Accesibility*”.

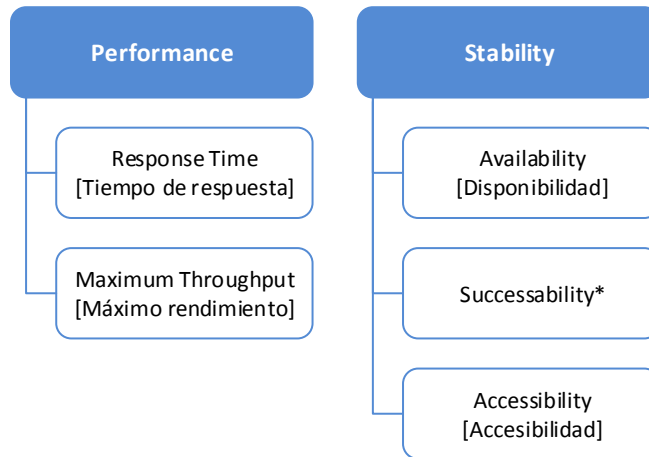


Fig. 6: Sub-factores de Calidad.

- **Response Time:** Se refiere a la duración del tiempo del enviar de una petición y el tiempo de recibir una respuesta. El tiempo de respuesta puede variar por el punto de medición y afectado por tres tipos de latencia: latencia del cliente, latencia de la red y latencia del servidor.
- **Maximum Throughput:** Se refiere a la cantidad máxima de servicios que el proveedor de servicio puede procesar en un periodo de tiempo dado. Esto es el máximo número de respuestas que pueden ser procesadas en una unidad de tiempo.
- **Availability:** Es una medición que representa el grado de cuales Servicios Web están disponibles en estado operacional. Esto se refiere a una proporción de tiempo en que el servidor del Servicio Web está activo y en ejecución. DownTime respresenta el tiempo cuando el servidor de un Servicio Web no está disponible para usar y UpTime representa el tiempo cuando el servidor está disponible, Disponibilidad (en inglés, Availability) se refiere al cociente de tiempo de actividad (en inglés, UpTime) entre el tiempo medido (en inglés, measured time). Para calcular la disponibilidad es conveniente usar el tiempo de inactividad (en inglés, DownTime) en lugar del tiempo de actividad.
- **Accessibility:** Representa la probabilidad de que una plataforma de Servicio Web es accesible mientras el sistema está disponible. Esto es una proporción de recepción de mensaje de confirmación (en inglés, receiving Ack message) de la plataforma al solicitar los servicios. Esto se expresa como el cociente del número de mensajes confirmados devueltos entre el número de mensajes de petición en un tiempo dado. Para incrementar la accesibilidad, un sistema necesita ser construido en una arquitectura extensible.
- **Successability:** Es una probabilidad de regreso de respuestas después de que Servicios Web son exitosamente procesados. En otras palabras, se refiere al cociente de número de mensajes respuestas entre el número de mensajes de solicitud después del procesamiento completo de servicios en un tiempo dado. "Successability" significa el caso de que un

mensaje de respuesta definido en WSDL es retornado. En este momento, se supone que un mensaje de petición es un mensaje de error.

Según [O'Brien, 2005] se definen los siguientes atributos de calidad:

- Interoperability
- Reliability
- Availability
- Usability
- Security
- Performance
- Scalability
- Extensibility
- Adaptability
- Testability
- Auditability
- Operability and Deployability
- Modifiability

4.7.2 Atributos de calidad & Métricas en selección de Servicios Web

En esta sección se presentan atributos de calidad de Servicios Web y sus respectivas métricas, así como una clasificación de atributos de calidad. Estas métricas están enfocadas a medir el nivel del servicio para servicios elementales o tareas dentro de un flujo de trabajo de un servicio compuesto. Para un mejor entendimiento de los atributos de calidad aquí presentados, diríjase a la sección 4.7.1, la cual contiene conceptos o definiciones de atributos de calidad.

A continuación se presentan un conjunto de tablas que presentan y describen atributos de calidad y métricas para WS. La columna Id en algunas tablas es para ubicar o hacer referencia de que parte de la fórmula se está describiendo, en este caso el símbolo.

Tabla 4: Atributos de calidad y Métricas [Lee 2010].

Atributo	Métrica
Response time	$ClientLatency = CL1 + CL2$ $NetworkLatency = NL1 + NL2$ $ServerLatency = SL1 + SL2 + SL3$ $ResponseTime = ClientLatency + NetworkLatency$
Máximum throughput	$\max \left(\frac{NumberofRequestsProcessedbyServiceProviderInMeasuredTime}{MeasuredTime} \right)$
Availability	$1 - \frac{DownTime}{MeasuredTime}$
Accessibility	$\frac{NumberofAckMessage}{NumberofRequestedMessage}$
Successability	$\frac{NumberofResponseMessage}{NumberofRequestedMessage}$

Tabla 5. Atributos de calidad y Métricas [Cardoso, 2004], [Zeng, 2003]. [Hwang, 2007]

Atributo	Métrica	Id
Response time	$T(t) = DT(t) + PT(t)$	1
Cost	$C(t) = EC(t) + RC(t)$	2
Reliability	$1 - (system\ failure\ rate + process\ failure\ rate)$	3
Notación		
Símbolo	Descripción	Id
DT	“Delay time”, se refiere al tiempo de no valor añadido necesario para que una instancia sea procesada por una tarea.	1
PT	“Process time”, es el tiempo que le lleva a una instancia de flujo de trabajo en una tarea mientras se está procesando; en otras palabras, corresponde con el tiempo que una tarea necesita para procesar una instancia.	1
EC	“enactment cost” es el costo asociado con la gestión de los sistemas de flujo de trabajo y con el seguimiento de las instancias del flujo de trabajo.	2
RC	“realization cost” es el costo asociado con la ejecución de la tarea del tiempo de ejecución.	2

Tabla 6. Atributos de calidad y Métricas [Yu, 2007]

Atributo de calidad	Métrica	Id
Response time	$Reaction\ time = T_2 - T_1$ $Receiving\ time = T_3 - T_2$ $Response\ time = Reaction\ time + Receiving\ time$	1
Throughput	$\frac{(\text{number of requests processed})}{(\text{unit time})}$	2
Reliability	$Ratio\ of\ failure\ in\ t\ days = n/N$ $Daily\ average\ failure\ rate = n/(N * t)$ $Success\ rate\ or\ Reliability = 1 - (n/(N * t))$	3
Scalability	$PNL(P, Q) = \sum Pr(s) C(s)$	4
Robustness	$MTBF = MTTF + MTTR$	5
Accuracy	Puede ser medido como la desviación estándar de “reliability”, pero no se especifica si es para una distribución de probabilidad continua o discreta.	6
Integrity	$\frac{(\text{number of successful transactions})}{(\text{total number of transactions})}$	7
Availability	$1 - \left[\frac{(\text{down time})}{(\text{unit time period measured})} \right]$ Ó $\frac{(\text{up time})}{(\text{unit time period measured})}$	8
Accessibility	$\frac{(\text{number of acknowledgements received})}{(\text{total number of requests sent})}$	9
Interoperability	$\frac{(\text{total number of environments the Web service runs})}{(\text{total number of possible environments that can be used})}$	10
Security	$\frac{(\text{total number of attacks detected})}{(\text{total number of attack simulations})}$	11
Notación		
Símbolo	Descripción	id
T_1	El cliente envía una petición al servidor Web.	1
T_2	El cliente empieza a recibir la respuesta del servidor.	1
$T_2 - T_1$	, Este intervalo es conocido como “reaction time” desde la perspectiva del servidor debido a que este es la cantidad de tiempo que este toma para que el servidor reaccione a las peticiones del cliente.	1
T_3	El cliente termina de recibir toda la respuesta enviada desde el servidor Web.	1
t	Número de días en que un WS es monitoreado para registrar el número de fallas.	3
n	Número de fallos que surgieron durante el período.	3
N	Es el número total de eventos (número de eventos con éxito más el número de fallos).	3
PNL	“The Performance Non-Scalability Likelihood” técnica para predecir si el sistema va a ser capaz de soportar las altas cargas de tráfico sin afectar a los niveles de desempeño.	4
$MTTF$	(Mean Time to Failure). Es el tiempo promedio que tarda el sistema en que falle una vez más.	5
$MTBF$	(Mean Time between Failures). Se puede definir como la cantidad de tiempo que toma para que el error se produzca, y luego recuperarse. Es mejor si el MTBF es más grande.	5
$MTTR$	(Mean Time to Recover). Es la cantidad media de tiempo que toma para que el sistema se recupere de un fallo.	5

(<i>down time</i>)	Es cuando el sistema no está disponible.	8
(<i>up time</i>)	Es cuando el sistema está disponible.	8

Tabla 7. Atributos de calidad y Métricas [Yu, 2005].

Atributo de calidad	Value
Response time (T)	$T = T_{sr} + T_{tr}$
Cost (C)	$C = C_{sr} + C_{tr}$
Service availability (A)	$A = T_a / T_t$
Service reliability (R)	$R = N_s / N$
Notación	
Símbolo	Descripción
T_{sr}	Tiempo de servicio.
T_{tr}	Tiempo de transmisión.
C_{sr}	Costo de servicio.
C_{tr}	Costo de transmisión.
T_a	Monto de tiempo que el servicio está disponible.
T_t	Tiempo total monitoreado.
N_s	Número de solicitudes respondidas completamente.
N	Total de solicitudes.

Tabla 8. Atributos de calidad y Métricas [Liu, 2004]

Atributo de calidad	Métrica
Execution Price	$q_{pr}(s)$
Execution duration	$q_{du}(s) = T_{process}(s) + T_{trans}(s)$
Reputation	$q_{rep} = \frac{\sum_{i=1}^n R_i}{n}$
Notación	
Símbolo	Descripción
s	El servicio a usar.
$T_{process}(s)$	Tiempo de procesamiento.
$T_{trans}(s)$	Tiempo de transmisión.
R_i	Es la valoración del usuario final sobre una reputación del servicio.
n	Es el número de veces que el servicio ha sido calificado.

Tabla 9. Atributos de calidad y Métricas [Patel, 2003].

Atributo de calidad	Métrica
Performance (Latency)	$t_{latency} = t_{o/p}(X) - t_{i/p}(X)$
Performance (Throughput)	$t_{throughput} = \#Service\ invocations\ in\ time\ T$
Reliability	$1 - P(succes)$ $P(succes) = \frac{\#successful\ executions}{\#TotalInvocations}$
Cost	$C(Enactment) + C(Licensing)$
Notación	
Símbolo	Descripción
$t_{i/p}(X)$	Es la marca de tiempo cuando el servicio X es invocado.
$t_{o/p}(X)$	Es la marca de tiempo cuando el servicio X es entregado.
$P(succes)$	Es la probabilidad de ejecuciones exitosas.

<i>C(Enactment)</i>	Costo de aprobación (gestión de flujo de trabajo y la supervisión).
<i>C(Licensing)</i>	Concesión de licencias.

Tabla 10. Atributo de calidad y Métricas [Kalepu, 2003].

Atributo de calidad	Métrica
“Local Compliance” de un atributo a^i	$L_{compl}^{a_i} = \sum_{j=1}^n \frac{Nd_j^{a_i}}{n}$
“Global compliance” de un atributo a^i	$GL_{compl}^{a_i} = \sum_{j=1}^n \frac{Nd_j^{a_i}}{n}$
“Local compliance” de un servicio	$WSL_{compl} = \frac{\sum_{i=1}^n L_{compl}^{a_i}}{n}$
“Global compliance” de un servicio	$WSG_{compl} = \frac{\sum_{i=1}^n GL_{compl}^{a_i}}{n}$
“Local compliance” de un proveedor de servicio	$SPL_{compl} = \frac{\sum_{i=1}^n WSL_{compl}^i}{n}$
“Global compliance” del proveedor del servicio	$SPG_{compl} = \frac{\sum_{i=1}^n WSG_{compl}^i}{n}$
“verity” de un servicio	$WSL_{verity} = \frac{\sum_{i=1}^n (WSL_i - \mu)^2}{n}$
“Global verity” de un servicio	$WSG_{verity} = \frac{\sum_{i=1}^n (WSG_i - \mu)^2}{n}$
“Local verity” de un proveedor o servicio	$SPL_{verity} = \frac{\sum_{i=1}^n (WSL_i - \mu)^2}{n}$
“Global verity” de un proveedor de servicio	$SPG_{verity} = \frac{\sum_{i=1}^n (WSG_i - \mu)^2}{n}$
“Local Reputation” de un WS	$\langle WSL_{compl}, WSL_{verity}, SL_{rating} \rangle$
“Global Reputation” de un WS	$\langle WSG_{compl}, WSG_{verity}, SL_{rating} \rangle$
“Local Reputation” de un proveedor de servicio	$\langle SPL_{compl}, SPL_{verity}, SPL_{rating} \rangle$

“Global Reputation” de un proveedor de servicio $\langle SPG_{compl}, SPG_{verity}, SPG_{rating} \rangle$

Tabla 11: Atributos de calidad y Métricas [Zeng, 2003],[Zeng, 2004].

Atributo de calidad	Métrica
Execution price	$q_{price}(s, op)$
Execution duration	$q_{du}(s, op)$ $q_{du}(s, op) = T_{process}(s, op) + T_{trans}(s, op)$ $T_{trans}(s, op) = \frac{\sum_{i=1}^n T_i(s, op)}{n}$
Reliability	$q_{rel}(s) = N_c(s)/K$
Availability	$q_{av}(s) = T_a(s)/0$
Reputation	$q_{rep} = \frac{\sum_{i=1}^n R_i}{n}$

Tabla 12: Atributos de calidad y Métricas [Rajendran 2009].

Atributo de calidad	Métrica
Integrity	$\frac{\text{Number of succesful transactions}}{\text{Total number de transactions}}$
Accessability	$\frac{\text{Number of acknowledgements received}}{\text{Total number of requests sent}}$
Interoperability	$\frac{\text{Total number of environments the Web Service runs}}{\text{Total number of possible environments that can be used}}$

Tabla 13: Atributos de calidad y Métricas [Zheng 2010].

Atributo de calidad	Métrica
failure probability of Web service i	$\bar{f}_i = \frac{1}{m} \sum_{a=1}^m f_{a,i}, \quad (1)$ $s_i = \sqrt{\frac{1}{m} \sum_{a=1}^m (f_{a,i} - \bar{f}_i)^2}, \quad (2)$
average failure probability of a service user a	$\bar{f}_a = \frac{1}{n} \sum_{i=1}^n f_{a,i}, \quad (3)$ $s_a = \sqrt{\frac{1}{n} \sum_{i=1}^n (f_{a,i} - \bar{f}_a)^2}, \quad (4)$
Tiempo de respuesta	$\frac{\text{tiempo de duracion del envio de una solicitud}}{\text{tiempo de duracion de la recepcion de la respuesta}}$
Rendimiento	Se define como la tasa media de tamaño de mensaje de éxito (aquí en bits) entregado a través de un canal de comunicación por segundo.
Notación	
Símbolo	Descripción
$f_{a,i}$	De la fórmula 1, es la probabilidad de falla del WS i observado por el usuario de servicio a .

m	De la fórmula 1, es el número de usuarios de servicio.
$\bar{f}_i$	De la fórmula 1, es el promedio de la probabilidad de falla del WS i . La desviación estándar de la probabilidad de falla del WS i se calcula en la fórmula 2.
$\bar{f}_i$	De la fórmula 2, es el promedio de la probabilidad de falla del WS i .
s_i	De la fórmula 2, es la desviación estándar de la probabilidad de falla del WS i .
n	De la fórmula 3, es el número de WS.
$\bar{f}_a$	De la fórmula 3, es la media del usuario del servicio a . la desviación estándar de la probabilidad de falla del usuario del servicio a es calculada en la fórmula 4.
$\bar{f}_a$	De la fórmula 4, es la media del usuario del servicio a .
s_a	De la fórmula 4, es la desviación estándar del usuario de servicio a .

Tabla 14: Atributos de calidad y métricas relacionadas con "performance" [Oberortner 2011], [Oberortner 2010], [Rajendran 2009] y [Rosenberg 2006].

Atributo de calidad	Métrica
Processing time	$t_p(S, o)$
Wrapping Time	$t_w(S, o)$
Execution Time	$t_e(S, o)$ $t_e = t_p + 2 * t_w$
Latency	$t_l(S)$
Response Time	$t_r(S, o)$ $t_r(S, o) = t_e(S, o) + 2 * t_l(S)$
Round Trip Time	$t_{rt} = (t_w)_{con.} + t_l + (t_p + 2 * t_w)_{provider} + t_l + (t_w)_{con.}$
Throughput	$tp(S, o) = \frac{\#R}{time\ period\ (in\ sec)}$
Scalability	$sc(S) = \frac{t_{rt}}{t_{rt}(Throughput)}$

Tabla 15: Atributos de calidad y métricas relacionadas con "Dependability" [Rosenberg 2006].

Atributo de calidad	Métrica
Availability	$av(S) = 1 - \frac{downtime}{uptime}$
Accuracy	$ac(S) = 1 - \frac{\#failed\ requests}{\#total\ requests}$
Robustness	$ro(S) = \frac{\sum_i^n f(resp_i(req_i(S)))}{\#total\ requests}$ $f = \begin{cases} 1, & isValid(resp_i) \\ 0, & \neg isValid(resp_i) \end{cases}$

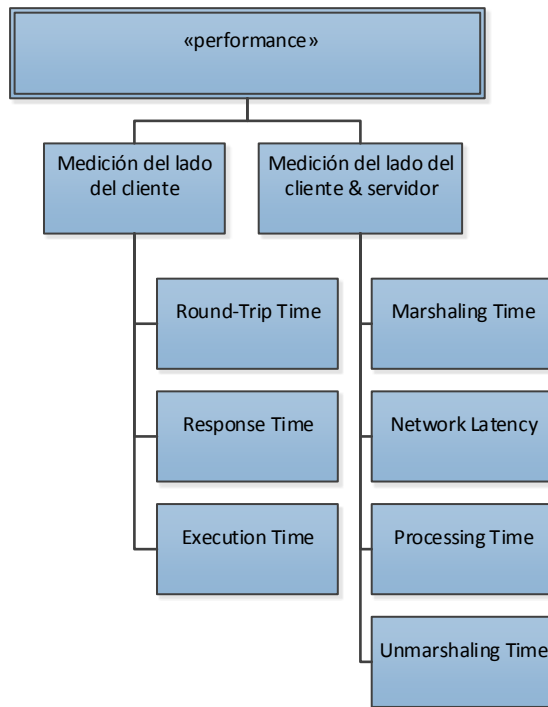


Fig. 7: Atributos de calidad relacionados con "performance" [Oberortner 2011].

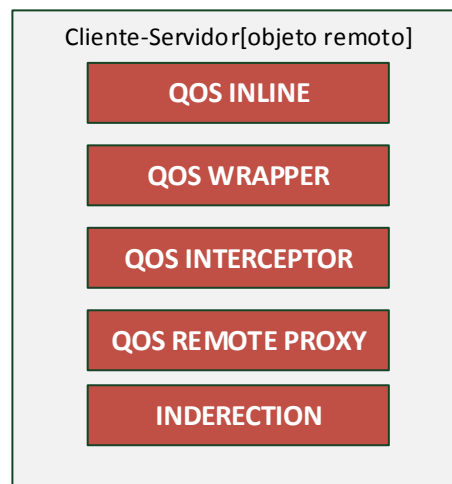


Fig. 8: Patrones para medición de atributos de calidad relacionados con "performance" [Oberortner 2011, Oberortner 2010, Rajendran 2009, Rosenberg 2006].

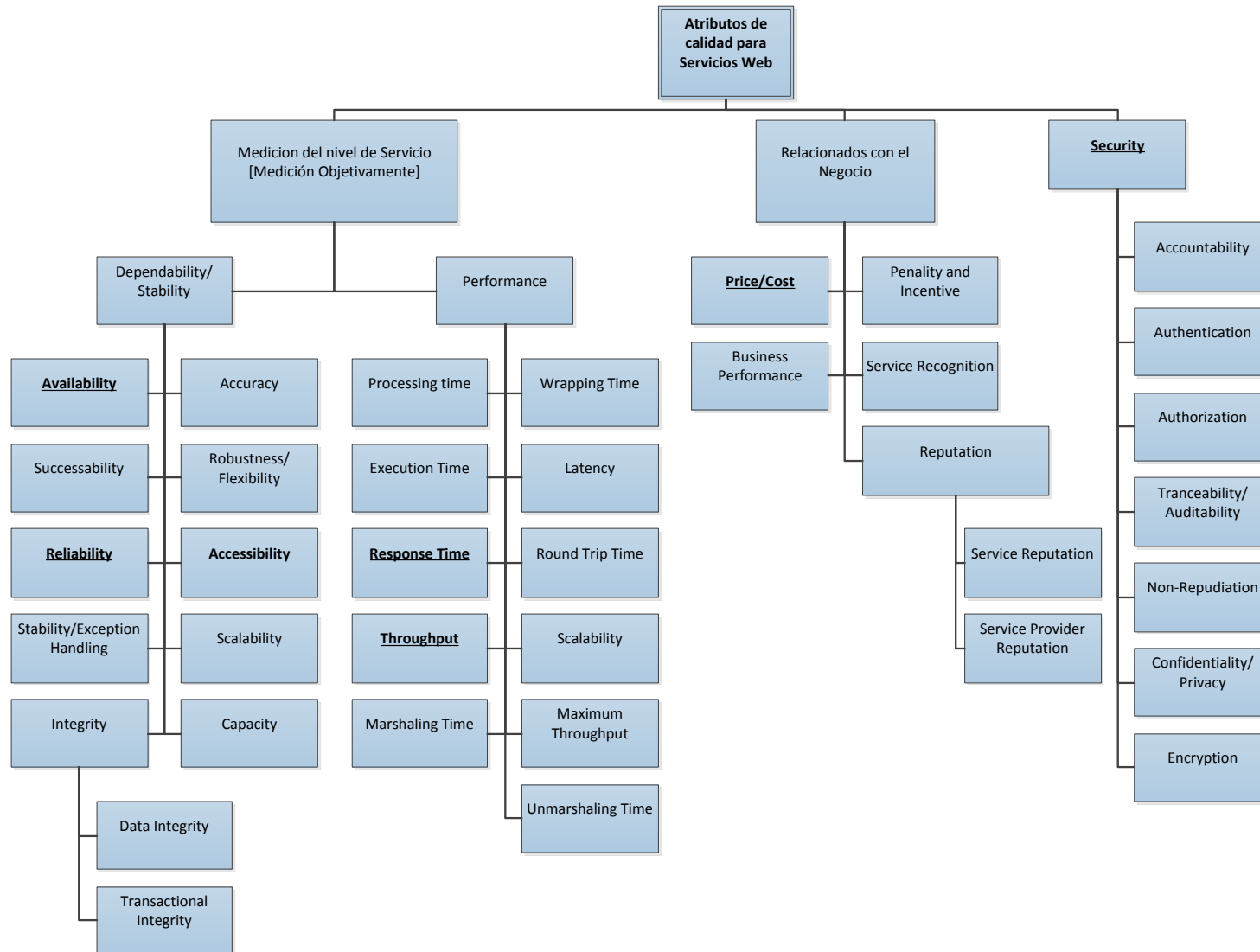


Fig. 9: Clasificación de atributos de calidad para WS.

Diferentes enfoques existen en la selección de Servicios Web, por ejemplo, dentro de los trabajos aquí estudiados, algunos proponen, mecanismos, marcos de trabajo o modelos de atributos de calidad que permitan la fácil selección, ejecución y enlace de Servicios Web, lo más común es usar agentes de selección de WS los cuales pueden soportar los requerimientos tanto funcionales como no funcionales de los usuarios.

En la selección de WS al usar agentes de selección de WS estos se auxilian de diferentes algoritmos que facilitan el cálculo de atributos de calidad en la selección, de forma que, el WS seleccionado sea el mejor u óptimo de un grupo de servicios. Aunque en algunos trabajos se menciona que se diseñan o usan tales algoritmos, no presentan implementaciones de tales algoritmos, solamente se menciona el nombre del algoritmo o algún ejemplo del algoritmo como tal.

Los atributos de calidad y métricas que se presentan de la tabla 4 a la tabla 15, tienen como objetivo el medir el nivel del servicio pero no presentan implementaciones de cómo medir excepto los atributos de calidad relacionados con “performance” de la tabla 14, los atributos de calidad de la tabla 14 pueden ser medidos tanto del lado del cliente como del lado del servidor, como se muestra en la figura 7, para tales mediciones, existen patrones para la medición de atributos de calidad relacionados con “performance”, como se muestra en la figura 8. Estos patrones de medición se encuentran en los libros [Buschmann, 1999],[Buschmann, 2007],[Gamma, 2004],[Schmidt, 2000],[Völter, 2005] que pueden ser de ayuda para arquitectos de software y desarrolladores que tienen que diseñar y desarrollar sistemas distribuidos y decidir cómo medir propiedades relacionadas con el rendimiento de la calidad de servicio dentro de sistemas distribuidos. Cabe mencionar que los patrones presentados no contemplan la manera de almacenar o evaluar las mediciones.

Algunos autores tales como [Wang 2007], [OASIS 2005] presentan, de alguna manera, clasificación de atributos de calidad para WS; tomando como base el trabajo de [Wang 2007] y habiendo estudiado los trabajos presentados en este trabajo, se presenta una clasificación de atributos de calidad para WS, tal como se ve en la figura 9. Dentro de esta clasificación se remarca con negritas y subrayado aquellos atributos de calidad que, según el autor [Diamadopoulou 2008] y observando en las tablas 4 a la 15, los de mayor demanda o de interés para el usuario o proveedor del servicio son: “*availability*”, “*security*”, “*throughput*”, “*reliability*”, “*price/cost*” y “*response time*”. Es importante remarcar que no todos los atributos de QoS de WS en esta clasificación cuentan con una métrica, en algunos casos, solamente se cuenta con definiciones o conceptos de tales atributos de calidad.

A continuación se presenta el concepto de cada uno de los atributos de calidad relacionados con “performance” con la finalidad de un mejor entendimiento.

Round-Trip Time: es una propiedad del lado del cliente y mide el tiempo transcurrido entre el envío de la petición del cliente y la recepción de la respuesta del servidor.

Marshaling Time: puede ser medida sobre el lado del cliente y del servidor. Es una medida del tiempo transcurrido para serializar los datos de invocación dentro en el formato de transporte de la red subyacente.

Network Latency: el tiempo requerido para transmitir los datos de invocación se calculan sobre la red se denomina latencia de red. Requiere puntos de medición en el cliente y el servidor. La latencia de la red puede ser medida durante la transmisión de la solicitud del cliente y durante la transmisión de la respuesta del servidor.

Response Time: Propiedad de QoS del lado del cliente y mide el tiempo transcurrido entre la transmisión de los datos de invocación serializados con el servidor y la recepción de la respuesta del servidor.

Processing Time: del lado del servidor, es el tiempo transcurrido para procesar una petición entrante. No se toma en cuenta: “Marshaling Time” y “Unmarshaling Time”.

Unmarshaling Time: se puede medir tanto del lado del cliente como del servidor. En el lado del servidor se mide el tiempo transcurrido de de-serializar los datos entrantes de invocación de la petición del cliente para que sea compatible con las capas superpuestas. Similar, en el lado del cliente es una medida del tiempo requerido de de-serializar los datos de invocación de la respuesta del servidor.

Execution Time: es una propiedad de QoS del lado del servidor. Es una medida sobre el tiempo total requerido de solicitud de un cliente, es decir, “unmarshaling”, “processing” y “marshaling”.

De la figura 9, la medición del nivel de servicio puede dividirse en “*Dependability*” y “*Performance*”. “*Performance*”, según [W3C 2003], [OASIS 2005] El rendimiento de un servicio web representa qué tan rápido se puede completar una solicitud de servicio. Puede medirse en términos de rendimiento, tiempo de respuesta, latencia, tiempo de ejecución y tiempo de la transacción y así sucesivamente.

“*Dependability/Stability*” según [OASIS 2005] La estabilidad significa qué tan estable y continua, los Servicios Web pueden proveer servicios. Es decir, la calidad se refiere a la capacidad de brindar un servicio continuo, consistente y recuperable a pesar de un mayor rendimiento, congestión, fallo del sistema, desastres naturales y ataques intencionales de los usuarios. Las propiedades de calidad son la disponibilidad, la fiabilidad y la accesibilidad, etc.

El valor de negocio de Servicios Web significa el valor económico entregado mediante la aplicación de Servicios Web en un negocio. El valor de negocio depende el precio de un servicio, una política de pena/remuneración, reconocimiento de servicio, reputación de servicio y reputación de proveedor de servicio, etc., [OASIS 2010] y [OASIS 2005].

Con respecto a “*Security*”, los autores [OASIS 2005] y [OASIS 2010], mencionan que la calidad de seguridad de Servicios Web es la capacidad para determinar la legalidad de acceso al sistema y servicio, para cortar cualquier ilegal ejercicio de enfoque, la fabricación y la autoridad, para controlar cualquier acceso legal y para proporcionar el servicio de seguridad integrado para el uso de autoridad estable, confiable y adecuado con el fin de reducir o eliminar todas las amenazas potenciales que pueden ocurrir durante el uso de Servicios Web.

4.7.3 Atributos de calidad & Métricas en Composición de Servicios Web

Tabla 16: Atributos de calidad y Métricas [Zeng, 2003]

atributos de calidad	Métrica
Execution price	$q_{price}(s, op)$
Execution duration	$q_{du}(s, op) \quad (1)$ $q_{du}(s, op) = T_{process}(s, op) + T_{trans}(s, op) \quad (2)$ $T_{trans}(s, op) = \frac{\sum_{i=1}^n T_i(s, op)}{n} \quad (3)$
Reliability	$q_{rel}(s) = N_c(s)/K \quad (4)$
Availability	$q_{av}(s) = T_a(s)/0 \quad (5)$
Reputation	$q_{rep} = \frac{\sum_{i=1}^n R_i}{n} \quad (6)$
Notación	
Símbolo	Descripción
op	Dada una operación.
s	Un servicio.
$T_{process}(s, op)$	De la fórmula 2. Es el tiempo de procesamiento.
$T_{trans}(s, op)$	De la fórmula 2. Es el tiempo de transmisión.
$T_i(s, op)$	De la fórmula 3. Es una observación pasada del tiempo de transmisión.
n	De la fórmula 3. Es el número de veces de ejecución observadas en el pasado.
$N_c(s)$	Es el número de veces que el servicio s ha sido completamente entregado dentro del marco de tiempo máximo esperado.
K	Es el número total de invocaciones.
T_a	Es el monto total de tiempo (en segundos) en que el servicio s está disponible durante los últimos 0 segundos (0 es una constante definida por un administrador de la comunidad de servicio).
R_i	Es la valoración del usuario final sobre la reputación de un servicio.
n	De la fórmula 6. Es el número de veces que el servicio ha sido calificado.

Tabla 17: Funciones de agregación para calcular QoS de WS compuestos [Zeng 2003].

Atributos de calidad	Función de agregación
Precio (Price)	$Q_{price}(p) = \sum_{i=1}^N q_{price}(S_i, op_i) \quad (1)$
Duración (Duration)	$Q_{du}(p) = CPA(q_{du}(S_1, op_1), \dots, q_{du}(S_N, op_N)) \quad (2)$
Reputación (Reputation)	$Q_{rep}(p) = \frac{1}{N} \sum_{i=1}^N q_{rep}(S_i) \quad (3)$
Confiabilidad (Reliability)	$Q_{rel}(p) = \prod_{i=1}^N (e^{q_{rel}(S_i) * z_i}) \quad (4)$
Disponibilidad (Availability)	$Q_{av}(p) = \prod_{i=1}^N (e^{q_{av}(S_i) * z_i}) \quad (5)$

Tabla 18: Atributos de calidad y métricas [Chen 2006].

Atributos de calidad	Métrica
ReponseTime	$CPA(Service_1, Service_2, \dots, Service_m)$

Availability	$\prod_{i=1}^m availability_i$
Concurrency	$\min(concurrency_1, concurrency_2, \dots, concurrency_m)$
ExpireTime	$\min(expireTime_1, expireTime_2, \dots, expireTime_m)$
Price	$\bigcup_{i=1}^m price_1$
Fine	A * Price, el cliente y el proveedor negocian para obtener A
securityLevel	$\min(securityLevel_1, securityLevel_2, \dots, securityLevel_m)$

Tabla 19: Atributos de calidad y funciones de agregación [Canfora 2005].

Atributos de calidad	Sequence	Switch	Flow	Loop
Time (T)	$\sum_{i=1}^m T(t_i)$	$\sum_{i=1}^n p_{ai} * T(t_i)$	$Max\{T(t_i)_{i \in \{1 \dots p\}}\}$	$K * T(t)$
Cost (C)	$\sum_{i=1}^m C(t_i)$	$\sum_{i=1}^n p_{ai} * C(t_i)$	$\sum_{i=1}^p C(t_i)$	$K * C(t)$
Availability (A)	$\prod_{i=1}^m A(t_i)$	$\sum_{i=1}^n p_{ai} * A(t_i)$	$A(t_i)$	$A(t)^k$
Reliability (R)	$\prod_{i=1}^m R(t_i)$	$\sum_{i=1}^n p_{ai} * R(t_i)$	$\prod_{i=1}^p R(t_i)$	$R(t)^k$
Custom Attr. (F)	$f_s(F(t_i))$ $i \in \{1 \dots m\}$	$f_B((P_{ai}, F(t_i)))$ $i \in \{1 \dots m\}$	$f_F(F(t_i))$ $i \in \{1 \dots m\}$	$f_L((k, F(t)))$
Notación				

Tabla 20: Atributos de calidad y funciones de agregación [You 2005].

Attributes	Aggregate Function (sequential)
Response time	$T = \sum_{i=1}^n (T_{si} + T_{ni})$
Cost	$C = \sum_{i=1}^n C_{si}$
Reliability	$R = \prod_{i=1}^n R_i$
Availability	$A = \prod_{i=1}^n A_i$

Tabla 21: Atributos de calidad y funciones de agregación [Zeng 2004].

Criterio	Función de agregación
Precio (Price)	$q_{pr}(p) = \sum_{i=1}^N q_{pr}(S_i, op(t_i))$

Duración (Duration)	$q_{du}(p) = CPA(p, q_{du})$
Reputación (Reputation)	$q_{rep}(p) = \frac{1}{N} \sum_{i=1}^N q_{rep}(S_i)$
Success rate	$q_{rat}(p) = \prod_{i=1}^N (q_{rat}(S_i)^{z_i})$
Disponibilidad (Availability)	$q_{av}(p) = \prod_{i=1}^N (q_{av}(S_i)^{z_i})$

Tabla 22: Atributos de calidad y métrica [Menascé 2002].

Attributes	Métrica
Desempeño	$X_{TA} \leq \min \left\{ \frac{X_a}{V_a}, \frac{X_b}{V_b}, \frac{X_c}{V_c} \right\}$

Tabla 23: Atributos de calidad y métrica [Menascé 2004].

Attributes	Métrica
Tiempo de ejecución	$T = t_A + p_1 t_B + p_2 (t_C + p_3 t_D + p_4 t_E) + t_F$
Costo total de la composición del WS	$C = c_A + p_1 c_B + p_2 (c_C + p_3 c_D + p_4 c_E) + c_F$

Tabla 24: Atributos de calidad y métricas [Li 2009].

Attributes	Métrica
Performance:	
• Throughput (Tp)	$Tp(t) = BN(t)/TI$
• Response Time (RT)	$RT(t) = ET(t) + DT(t)$
• Reliability (Re)	$Re(t) = Rli(t)/TI(t)$
• Availability (Av)	$Av(t) = Rln(t)/TI(t)$
Capacity:	
• Simultaneous Requests (SR)	$SR(t) = MSRN(t)/TI$

Tabla 25: Atributos de calidad y funciones de agregación [Cardoso 2004].

Atributos de calidad	Secuencial	Paralelo	Condicional
Response time	$T(t_{ij}) = T(t_i) + T(t_j)$	$T(t_{1n}) = \max_{i \in \{1..n\}} \{T(t_i)\}$	$T(t_{1n}) = \sum_{1 \leq i \leq n} p_{ai} * T(t_i)$

Cost	$C(t_{ij}) = C(t_i) + C(t_j)$	$C(t_{1n}) = \sum_{1 \leq i \leq n} C(t_i)$	$C(t_{1n}) = \sum_{1 \leq i \leq n} p_{ai} * C(t_i)$
Reliability	$R(t_{ij}) = R(t_i) + R(t_j)$	$R(t_{1n}) = \prod_{1 \leq i \leq n} R(t_i)$	$R(t_{1n}) = \sum_{1 \leq i \leq n} p_{ai} * R(t_i)$

Tabla 26: Atributos de calidad y funciones de Agregación [Cardoso 2004].

Atributos de calidad	Ciclo Simple	Ciclo Dual
Response time	$T(t_{ii}) = \frac{T(t_i)}{1 - p_i}$	$T(t_{ij}) = \frac{T(t_i) + T(t_j) - (1 - p_j)T(t_j)}{(1 - p_j)}$
Cost	$C(t_{ii}) = \frac{C(t_i)}{1 - p_i}$	$C(t_{ij}) = \frac{C(t_i) + C(t_j) - (1 - p_j)C(t_j)}{(1 - p_j)}$
Reliability	$R(t_{ii}) = \frac{(1 - p_i) * R(t_i)}{1 - p_i R(t_i)}$	$C(t_{ij}) = \frac{(1 - p_i) * R(t_i)}{1 - p_i R(t_i) R(t_j)}$

Tabla 27: Atributos de calidad y funciones de agregación [Cardoso 2004].

Atributos de calidad	Tolerante a fallos
Response time	$T(t_{1n}) = \min_k(\{T(t_1), \dots, T(t_n)\})$
Cost	$C(t_{1n}) = \sum_{1 \leq i \leq n} C(t_i)$
Reliability	$R(t_{1n}) = \sum_{i_1=0}^1 \dots \sum_{i_n=0}^1 g\left(\sum_{j=1}^n i_j - k\right) * ((1 - i_1) + (2i_1 - 1)R(t_1)) * \dots * ((1 - i_n) + (2i_n - 1)R(t_n))$

Para la tabla 17, se tiene una breve descripción de cada función de agregación:

Execution Price: El precio de ejecución $Q_{price}(p)$ de un plan p es una suma del precio de ejecución $q_{price}(S_i, op_i)$ de cada servicio S_i .

Execution duration: La duración de ejecución $Q_{du}(p)$ de un plan de ejecución p es calculada usando el Algoritmo de Ruta Critica CPA. Específicamente el CPA es aplicado a la ruta de ejecución del plan p .

Reputation: La reputación $Q_{rep}(p)$ de una ejecución del plan p es el promedio de la reputación $Q_{rep}(p)$ de cada servicio S_i en el plan de ejecución p .

Reliability: La confiabilidad $Q_{rel}(p)$ de un plan de ejecución p es un producto de $e^{q_{rel}(s_i) \cdot z_i}$. En la función de agregación, z_i es igual a 1 si el servicio es crítico sino, entonces es igual a 0 y $e^{q_{rel}(s_i) \cdot z_i} = 1$.

Availability: la Disponibilidad de un plan de ejecución es un producto de $e^{q_{av}(s_i) \cdot z_i}$, donde $q_{av}(s_i)$ es la disponibilidad del servicio.

Usando las funciones de agregación anterior, la calidad de vector de un plan de ejecución de servicios compuestos es definido como:

$$Q(p) = (Q_{price}(p), Q_{du}(p), Q_{rep}(p), Q_{rel}(p), Q_{av}(p))$$

Para la tabla 18 se tiene las siguientes definiciones de atributos de calidad:

ResponseTime: representa el tiempo de respuesta de un elemento del servicio.

Availability: representa la probabilidad de que el servicio pueda ser usado correctamente.

Concurrency: representa la habilidad máxima para soportar transacciones concurrentes.

ExpireTime: representa la caducidad de un servicio, y la confiabilidad de un servicio puede ser asegurada antes de expireTime.

Price: representa el dinero del cliente que deberá pagar por su servicio.

Fine: representa el servicio proveedor-cliente debería compensar cliente-proveedor para romper el contrato entre ellos. Comúnmente, fine tiene una relación linear con el precio.

Security Level: representa el nivel de seguridad de un servicio.

ResponseTime: considerando que existen servicios concurrentes en Servicios Web compuestos, el tiempo de respuesta del servicio compuesto no es la suma del tiempo de respuesta de todos los elementos del servicio, esto debería ser la suma de la ruta crítica en proceso de ejecución. CPA es un algoritmo para encontrar la ruta crítica.

En la composición de WS, esta puede ser llevada a cabo mediante patrones de composición abstracta o también llamados flujos de composición, tales como: en paralelo, secuencia, ciclos, condicional; para cada flujo de composición de servicios existen funciones de agregación de calidad, las cuales, permiten determinar la calidad de un servicio compuesto mediante la agregación de QoS de los servicios individuales que lo componen. En las siguientes

tablas de la 16 a la 29 contienen funciones de agregación; en las tablas 11, 14, 16 y 24 contienen métricas para servicios individuales, por ejemplo para representar el precio de un servicio individual, se expresa de la siguiente forma:

$$q_{price}(s, op),$$

Cuando esta forma parte de un servicio compuesto, para calcular el precio de tal composición se representa de la siguiente forma:

$$Q_{price}(p) = \sum_{i=1}^N q_{price}(S_i, op_i)$$

En otras palabras, es la sumatoria del precio de todos los servicios que componen el flujo de servicios.

En las tablas 28 y 29 se puede apreciar el tipo de agregación, por ejemplo, sumatorias, mínimo, multiplicación o producto; así como el tipo de unidad en la que se pueden medir estos atributos de calidad.

Por otra parte, los diferentes tipos de flujos de composición, el más común en uso, es el flujo de composición secuencial, aunque el autor [Jang 2006] afirma que, los flujos de condicional, paralelo, etc, pueden convertirse al flujo secuencial mediante técnicas de transformación.

Con respecto a la composición en la nube computacional, diferentes enfoques se presentan para la selección de servicios, así como también para resolver problemas de optimización en la composición de WS. Algunos conceptos sobre composición de servicios se mencionan a continuación: un servicio compuesto abstracto puede ser definido como una representación abstracta de una petición de composición:

$$SC = \{S_1, \dots, S_N\}$$

$SC = \text{Service Classes}$

Donde SC se refiere a las clases de servicio requeridas. Un servicio compuesto concreto puede ser definido como una instanciación de un servicio compuesto abstracto. Un servicio compuesto concreto puede ser obtenido mediante el enlace de cada clase de servicio en SC para un servicio concreto s_j , donde $s_j \in S_j$ y $S_j = \{s_{j1}, \dots, s_{jl}\}$ contienen $l (l > 1)$ servicios de funcionalidad equivalente con diferentes valores de QoS.

Las métricas que a continuación se mencionaran están presentes en los siguientes trabajos: [Alrifai, 2009],[Alrifai, 2010],[Ardagna, 2007],[Cui, 2012],[Falasi, 2011],[Klein, 2012],[Qi, 2011],[Rosenberg, 2009],[Wang, 2011],[Yu, 2007], [Jaeger, 2004], [Jaeger, 2005], [Jaeger, 2006] comúnmente son llamadas funciones de agregación, pues la calidad de un Servicio Web compuesto depende de cada uno de los servicios elementales que lo componen. Cabe señalar que en composición de Servicios Web existen diferentes modelos de composición o también llamados modelos de flujo de trabajo como lo son: secuencial, ciclo, paralelo, condicional, entre otros; el más común usado es el secuencial, para cada modelo de composición las métricas son diferentes en su expresión matemática; pero los modelos diferentes a secuencial pueden ser transformados al modelo secuencial usando técnicas de transformación según [Jang, 2006].

Los problemas de optimización en composición de servicios comúnmente o se puede ver la tendencia en usar MIP “Mixed Integer Programming” para elegir los mejores candidatos para la composición. En MIP son usadas las funciones de utilidad para mapear el vector de valores de atributos de calidad dentro de un solo valor real para permitir la ordenación y valoración de los candidatos de servicio. Por ejemplo el mínimo y máximos valores agregados de k-ésimo atributo de QoS de S son calculados como sigue:

$$U(s) = \sum_{k=1}^r \frac{Q_{j,k}^{max} - q_k(s)}{Q_{j,k}^{max} - Q_{j,k}^{min}} \cdot W_k$$

$$U(S) = \sum_{k=1}^r \frac{Q_{j,k}^{max} - q_k(S)}{Q_{j,k}^{max} - Q_{j,k}^{min}} \cdot W_k$$

$$Q_k^{max} = \sum_{j=1}^n Q_{j,k}^{max}, Q_{j,k}^{max} = \max_{s_{ji} \in S_j} q_k(s_{ji})$$

Con

$$Q_k^{min} = \sum_{j=1}^n Q_{j,k}^{min}, Q_{j,k}^{min} = \min_{s_{ji} \in S_j} q_k(s_{ji})$$

Donde $W_k \in R^+$ ($\sum_{k=1}^r W_k = 1$) representa las preferencias de los usuarios, $Q_{j,k}^{min}$ es el valor mínimo de los k-ésimos atributo en todas las clases candidatas de servicio S_j y de manera similar $Q_{j,k}^{max}$ es el valor máximo, $Q_{j,k}^{min}$ es el valor mínimo de los k-ésimos atributos de S y de manera similar Q_k^{max} es el valor máximo.

La selección de servicio con limitantes de QoS globales es un proceso de optimización. La selección óptima para una composición de servicio dada S debe cumplir con las siguientes condiciones.

Para un vector dado de limitantes de QoS globales $CS = \{C_1, \dots, C_m\}$ ($0 \leq m \leq r$), $q(S) \leq C$ ($\forall C_K \in CS$), donde $q(S)$ es el valor de QoS agregado de la composición de servicio.

La utilidad total del máximo valor de $U(S)$ en la composición de servicio.

Sin embargo, encontrar la composición óptima requiere enumerar todas las posibles combinaciones de los candidatos de servicio, que pueden ser muy caros en términos de tiempo de cálculo. Por lo tanto, es difícil asegurar la confiabilidad de los servicios seleccionados.

A continuación en las tablas 28 y 29 se presentan las funciones de agregación con mayor tendencia en uso:

Tabla 28: Atributos de calidad y funciones de agregación.

Aggregation type	Atributos de calidad	Sequential
Summation	Price, Response time	$q(CS) = \sum_{j=1}^n q(s_j)$
Minimum	Throughput	$q(CS) = \min_{j=1}^n q(s_j)$
Summation	Reputation	$q(CS) = \frac{1}{n} \sum_{j=1}^n q(s_j)$
Multiplication	Availability, Reliability	$q(CS) = \prod_{j=1}^n q(s_j)$

Tabla 29: Atributos de calidad y funciones de agregación.

Attribute	Unit	Sequence	Conditional (XOR)	Parallel (AND)	Loop
Response Time q_{rt}	msec	$\sum_{i=1}^n q_{rt}(f_i)$	$\sum_{i=1}^n p_i$ $* q_{rt}(f_i)$	$\max\{f_1, \dots, f_n\}$	$q_{rt}(f)$ $* c$
Latency q_l	msec	$\sum_{i=1}^n q_l(f_i)$	$\sum_{i=1}^n p_i$ $* q_l(f_i)$	$\max\{f_1, \dots, f_n\}$	$q_l(f)$ $* c$
Price q_p	per invocation	$\sum_{i=1}^n q_p(f_i)$	$\sum_{i=1}^n p_i$ $* q_p(f_i)$	$\sum_{i=1}^n q_p(f_i)$	$q_p(f)$ $* c$
Availability q_{av}	percent	$\prod_{i=1}^n q_{av}(f_i)$	$\sum_{i=1}^n p_i$ $* q_{av}(f_i)$	$\prod_{i=1}^n q_{av}(f_i)$	$q_{av}(f)^c$

Accuracy q_{ac}	percent	$\prod_{i=1}^n q_{ac}(f_i)$	$\sum_{i=1}^n p_i$ $* q_{ac}(f_i)$	$\prod_{i=1}^n q_{ac}(f_i)$	$q_{ac}(f)^c$
Throughput q_{tp}	invocation/sec	$\min \{f_1, \dots, f_n\}$	$\sum_{i=1}^n p_i$ $* q_{tp}(f_i)$	$\min \{f_1, \dots, f_n\}$	$q_{tp}(f)$
Reliable Messaging q_{rm}	{true, false}	$q'_{rm} = \begin{cases} true & \forall 1 < i \leq n \ q_{rm}(f_i) = true \\ false & \exists f_i \in F \ q_{rm}(f_i) = false \end{cases}$			
Security q_{sec}	{None,X.509,. . .}	$q'_{sec} = \begin{cases} X.509 & \forall 1 < i \leq n \ q_{sec}(f_i) = X.509 \\ None & otherwise \end{cases}$			

CAPÍTULO 5: Conclusión y trabajos futuros

En este capítulo se describen las conclusiones del desarrollo de la tesis. Así como también se presentan los trabajos futuros resultantes de esta investigación.

5.1 Conclusiones

En este trabajo de investigación se aborda el problema de que en los procesos de construcción, selección y composición de Servicios Web se propone el uso o empleo de atributos de calidad como base para estos procesos. Sin embargo no se tiene identificado como medir esos atributos de calidad, en consecuencia, los valores de los atributos de calidad son arbitrarios en muchos de los procesos.

Atendiendo a esta problemática, la aportación de esta tesis es un documento del estado del arte el cual consiste en identificar un conjunto de atributos de calidad y un conjunto de fórmulas, con las que se puedan obtener datos que puedan ser utilizados como marco de valores de calidad, para un atributo de calidad cuando se construya, seleccione o se requiera que un Servicio Web participe en un proceso de composición. La importancia de incluir o gestionar atributos de calidad en los Servicios Web es motivada por dos partes, la primera, se refiere al consumidor del Servicio Web por experimentar el buen desempeño del servicio, la otra se refiere a que los proveedores necesitan formular ofertas compatibles con atributos de calidad a fin de obtener el mayor beneficio del servicio, por lo que es importante para que los usuarios y proveedores de Servicios Web puedan entender a los atributos de calidad y sus formas de medirlos. Cabe mencionar que cuando se construye un nuevo servicio, este puede ser desarrollado a través del proceso de composición de WS.

Debido a lo anterior se concluye que el objetivo de esta tesis se cumple, pues se presentan atributos de calidad y métricas para los dominios de selección y composición. Para el logro del objetivo se presenta una taxonomía de atributos de calidad que básicamente se divide en los dominios de selección, composición y reutilización, facilitando así ubicar el atributo de calidad y su métrica dentro de un dominio determinado. Es importante mencionar que se presenta como resultado del estudio de los trabajos seleccionados una clasificación de atributos de calidad para WS, la cual se divide en medición del nivel de servicio, relacionados con el negocio y seguridad. Dentro de esta clasificación se remarca con negritas y subrayado aquellos atributos de calidad que son los de mayor demanda o de interés para el usuario o proveedor.

Los alcances se cumplen ya que solamente se estudiaron atributos de calidad de WS, así como, sus respectivas métricas, con respecto a las limitaciones se concluye que aunque la cantidad de atributos de calidad no es exhaustiva, en esta tesis se presenta una cantidad considerable de atributos de calidad.

Una evolución de los WS es la migración hacia la nube computacional, pues para los proveedores de servicios y empresas de negocios, representa grandes beneficios, tales como el no

tener una inversión en infraestructura inicial y el sólo pagar por el servicio consumido, así como, pagar licencias de software y gastos de mantenimiento tanto de hardware y software. Algunos problemas con los servicios en la nube necesitan ser atendidos, como son:

- La seguridad y privacidad de datos, pues uno de los obstáculos en la migración hacia la nube, es precisamente esa incertidumbre sobre eso.
- Debido a que el estándar de UDDI aún no ha sido ampliamente desplegado o adoptado en la nube, se requiere de más investigación para facilitar los procesos de descubrimiento de servicios en este entorno.
- Los proveedores deberían de diseñar SLA's (visto en el punto 4.2.5 de esta tesis) de diferente tipo, por ejemplo: servicio gold, servicio silver, servicio bronce, prestando así un servicio flexible al cliente.

5.2 Trabajos futuros

Algunos trabajos futuros que se proponen al término de este trabajo se describen como sigue:

- Diseño, desarrollo e implementación de un banco de pruebas para evaluar a las métricas de calidad obteniendo resultados que pueden ayudar a decidir cuál métrica es la que arroja un valor representativo del servicio, la implementación puede ser en una red local, web o en la nube.
- Diseño, desarrollo e implementación de un sistema de monitoreo de QoS para sistemas orientados a servicios, con la finalidad de vigilar el cumplimiento de los SLA's entre proveedor y cliente.

Anexo A: Modelos de Calidad de Software

En este apartado se presenta Modelos de calidad de software.

A.1 Modelos de calidad de software

En este apartado se presenta el concepto de calidad de software y se dan a conocer algunos modelos, considerando sólo los más conocidos.

Para abordar el tema de calidad de software primeramente se dará a conocer el concepto de calidad de software, por lo tanto, se tiene lo siguiente:

- En [Pressman, 2002] tenemos que es la concordancia con los requisitos funcionales y de rendimiento explícitamente establecidos con los estándares de desarrollo explícitamente documentados y con las características implícitas que se espera de todo software desarrollado profesionalmente.
- Según la [ISO8402, 1994] es el conjunto de características de una entidad que le confieren su aptitud para satisfacer las necesidades expresadas y las implícitas.

Por otra parte con respecto a modelos de calidad de software en [Moreno, 2010] se menciona que *“un modelo completo se considera que posee descripción de atributos, métricas y mecanismos de ayuda para llegar a la medición”*.

Se han propuesto cientos de métricas para el software, pero no todas proporcionan un soporte práctico para el desarrollador de software. Algunas demandan mediciones que son demasiado complejos, otras son tan esotéricas que pocos profesionales tienen la esperanza de entenderlas y otras violan las nociones básicas intuitivas de lo que realmente es el software de alta calidad [Pressman, 2002].

En [Ejiogu, 1991] se define un conjunto de atributos que deberían acompañar a las métricas efectivas del software. La métrica obtenida y las medidas que conducen a ello deberían ser:

- *Simples y fáciles de calcular.* Debería ser relativamente fácil aprender a obtener la métrica y su cálculo no debería demandar un esfuerzo o cantidad de tiempo inusuales.
- *Empírica e intuitivamente persuasivas.* La métrica debería satisfacer las nociones intuitivas del ingeniero sobre el atributo del producto en cuestión (por ejemplo, una métrica que mide la cohesión de un módulo debería aumentar su valor a medida que crece el nivel de cohesión).

- *Consistentes y objetivas.* La métrica debería siempre producir resultados sin ambigüedad. Un tercer equipo debería ser capaz de obtener el mismo valor de métrica usando la misma información del software.
- *Consistentes en el empleo de unidades y tamaños.* El cálculo matemático de la métrica debería emplear medidas que no conduzcan a extrañas combinaciones de unidades. Por ejemplo, multiplicando el número de personas de un equipo por las variables del lenguaje de programación en el programa resulta una sospechosa mezcla de unidades que no son intuitivamente persuasivas.
- *Independientes del lenguaje de programación.* Las métricas deberían basarse en el modelo de análisis, modelo de diseño o en la propia estructura del programa. No deberían depender de los caprichos de la sintaxis o semántica del lenguaje de programación.
- *Un eficaz mecanismo para la realimentación de calidad.* La métrica debería proporcionar, al desarrollador de software, información que le lleve a un producto final de mayor calidad.

En los procesos de desarrollo de software de calidad, existen diferentes modelos de calidad de software, los cuales están formados por Atributos de calidad y sus respectivas métricas con el objetivo de medir la calidad del software, por ejemplo, si se deseara medir el atributo de calidad de “funcionalidad” podríamos usar el modelo ISO/IEC 9126. A continuación se mencionan algunos:

- Modelo FURPS [Moreno, 2007]
- Modelo ISO/IEC 9126 [Moreno, 2007]
- Modelo de Dromey [Moreno, 2010]
- Modelo QUINT2 – 2002 [Moreno, 2010]

Por mencionar el más relevante, el modelo ISO/IEC 9126:

En el trabajo de [Moreno, 2007], la organización internacional de estándares presentó su modelo ISO-9126, con gran aceptación entre la comunidad de ingeniería del software. El estándar se basa en los modelos de McCall y Boehm. Éste es un modelo jerárquico, que descompone la noción de calidad en seis factores (Funcionalidad, Usabilidad, Mantenibilidad, Confiabilidad, Portabilidad y Eficiencia). A la fecha, existen 3 sub estándares a partir de ISO-9126. El ISO-9126-2 define métricas externas, ISO-9126-3 define métricas internas vistas en la fig. 4, y el ISO-9126-4 define lineamientos para usar las métricas en la medición de las características y/o sub-características del modelo vistas en la fig. 5. Las métricas internas valoran al software

mismo, mientras que las métricas externas valoran el comportamiento del software en el contexto en el que se halla. Si bien el modelo es muy completo, los estándares aún no están plenamente especificados y la información es privada.

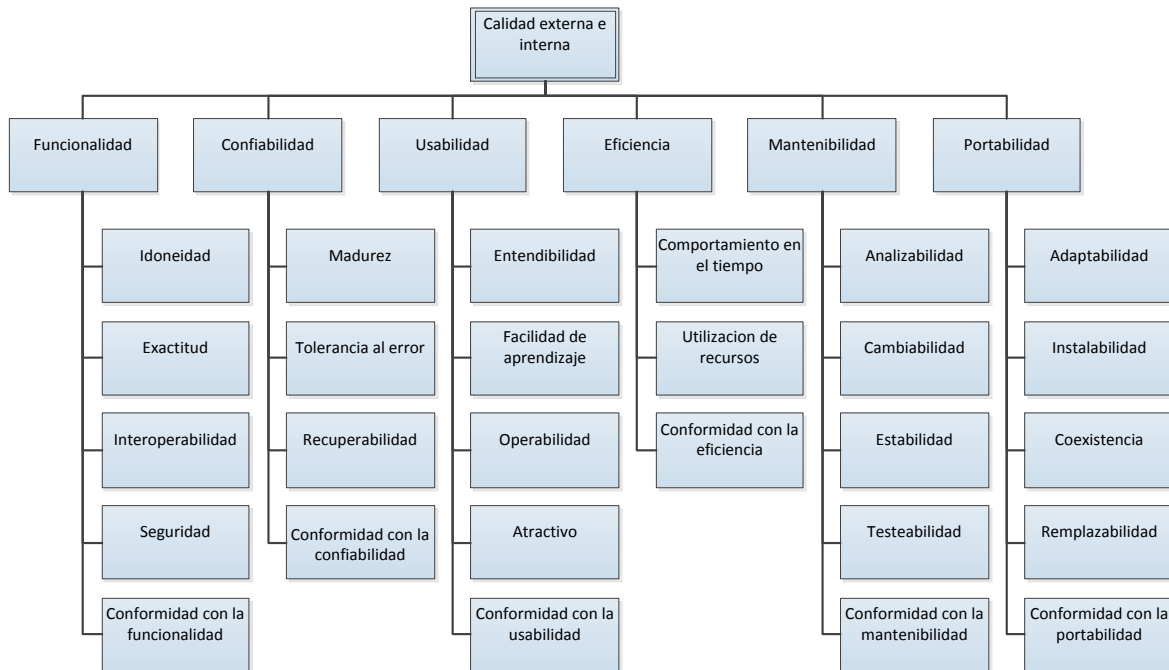


Fig. 10: Modelo de calidad para calidad Interna y Externa [Moreno, 2007].

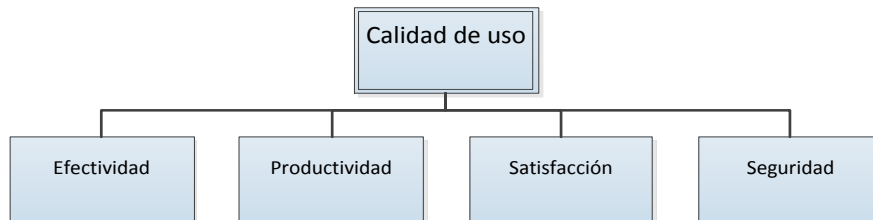


Fig. 11: Modelo de calidad para calidad de uso [Moreno, 2007].

Según el estándar ISO 9126, el cual se compone como se muestra en la fig. 4 y 5, la Funcionalidad es el grado en que el software satisface las necesidades indicadas por los siguientes subatributos: idoneidad, exactitud, interoperatividad, conformidad y seguridad.

Los factores ISO 9126 no necesariamente son utilizados para medidas directas. En cualquier caso, facilitan una valiosa base para medidas indirectas y una excelente lista para determinar la calidad de un sistema.

Sin embargo los atributos de calidad de software de los modelos antes mencionados pueden no ser aplicables en el ámbito de Servicios Web, para ello OASIS propone un modelo de calidad para Servicios Web, lo cual, se podrá hallar a detalle en el punto 4.6.2 de este trabajo.

Referencias

- [W3Schools, 2013] W3Schools. "WSDL y UDDI." http://www.w3schools.com/wSDL/wSDL_uddi.asp.
- [Wikipedia, 2013] Wikipedia, Colaboradores de. "Acuerdo De Nivel De Servicio." Wikipedia, La enciclopedia libre, http://es.wikipedia.org/w/index.php?title=Acuerdo_de_nivel_de_servicio&oldid=63784250.
- [Cui, 2012] Cui, Lizhen, Jian Li, and Yongqing Zheng. "A Dynamic Web Service Composition Method Based on Viterbi Algorithm " In 19th International Conference on Web Services, 267-71: IEEE Computer Society, 2012.
- [Hashemi, 2012] Hashemi, Seyyed Mohsen, and Amid Khatibi Bardsiri. "Cloud Computing Vs. Grid Computing." ARPN Journal of Systems and Software 2, no. 5 (mayo 2012): 188-94.
- [Klein, 2012] Klein, Adrian, Fuyuki Ishikawa, and Shinichi Honiden. "Towards Network-Aware Service Composition in the Cloud." WWW 2012 – Session: Web Engineering 2 (April 16–20 2012): 959-68.
- [Mendhe, 2012] Mendhe, Tushar Kailas, .P.A. Kamble, and Ashish K. Thakre. "Survey on Security, Storage, and Networking of Cloud Computing ". International Journal on Computer Science and Engineering (IJCSE) 4, no. 11 (2012).
- [Patidar, 2012] Patidar, S. , D. Rane, and P. Jain. "A Survey Paper on Cloud Computing." In Second International Conference on Advanced Computing & Communication Technologies (ACCT), 394-98: IEEE, 2012.
- [Valenzuela, 2012] Valenzuela, Blanca Dina Robles. "Marco Orientado a Objetos Para Cálculos De Similitud." CENIDET, 2012.
- [W3C, 2012b] W3C. "Groups." W3C, http://www.w3.org/Consortium/activities#Web_Services_Resource_Access_Working_Group.

- [W3C, 2012e] W3C. "Groups." W3C, <http://www.w3.org/Consortium/activities#about>.
- [W3C, 2012a] W3C. "Groups." W3C, http://www.w3.org/Consortium/activities#Web_Services_Policy_Working_Group.
- [W3C, 2012d] W3C. "Sobre El W3c." W3C españa, <http://www.w3c.es/Consortio/>.
- [W3C, 2012c] W3C. "Web Services Activity Statement." W3C, <http://www.w3.org/2002/ws/Activity.html>.
- [Falasi, 2011] Falasi, Asma Al, and Mohamed Adel Serhani. "A Framework for Sla-Based Cloud Services Verification and Composition ". Innovations in Information Technology (IIT), 2011 International Conference on (25-27 April 2011): 287-92.
- [González, 2011] González, Guadalupe Ariana Sánchez. "Evaluación De Medidas De Similitud Aplicadas a La Selección De Servicios Web ", cenidet, 2011.
- [Mell, 2011] Mell, Peter, and Timothy Grance. "The Nist Definition of Cloud Computing (Draft) ": National Institute of Standards and Technology (NIST) U.S. Departament of Commerce, 2011.
- [Oberortner, 2011] Oberortner, E, S Sobernig, Uwe Zdun, and S Dustdar. "Monitoring of Performance-Related Qos Properties in Service-Oriented Systems: A Pattern-Based Architectural Decision Model." (2011).
- [Pearson, 2011] Pearson, Siani. " Towards Accountability in the Cloud." IEEE Internet Computing 15, no. 4 (July-Aug 2011): 64-69.
- [Qi, 2011] Qi, Lianyong, Wanchun Dou, Xuyun Zhang, and Jinjun Chen. "A Qos-Aware Composition Method Supporting Cross-Platform Service Invocation in Cloud Environment." Journal of Computer and System Sciences 78, no. 5 (2011): 1316–29.
- [Wang, 2011] Wang, Shangguang, Zibin Zheng, Qibo Sun, Hua Zou, and Fangchun Yang. "Cloud Model for Service Selection." In Computer Communications Workshops (INFOCOM WKSHPS), 666-71. Shanghai: IEEE, 2011.

- [Alrifai, 2010] Alrifai, Mohammad, Dimitrios Skoutas, and Thomas Risse. "Selecting Skyline Services for Qos-Based Web Service Composition." WWW2010 Proceedings of the 19th international conference on World wide web (April 26-30 2010): 11-20.
- [Chen, 2010] Chen, Hao-peng, and Shao-chong Li. "Src: A Service Registry on Cloud Providin G Behavior-Aware and Qos-Aware Service Discovery." In SOCA, 2010, 1-4, 2010.
- [Goscinski, 2010] Goscinski, Andrzej, and Michael Brock. "Toward Dynamic and Attribute Based Publication, Discovery and Selection for Cloud Computing." Future Generation Computer Systems 26, no. 7 (2010): 947-70.
- [Kaufman, 2010] Kaufman, L.M. , and Bruce Potter. "Can Public-Cloud Security Meet Its Unique Challenges?". Security & Privacy, IEEE 8, no. 4 (2010): 55-57.
- [Lee, 2010] Lee, Youngkon. "Qos Metrics for Service Level Measurement for Soa Environment." In 6h International Conference on Advanced Information Management and Service (IMS), 509-14 Seoul: IEEE, 2010.
- [Liu, 2010] Liu, Rongjie, Guangsheng Cao, Jie Zhang, Pingjian Song, and Binge Cui. "Research on Dynamic Web Services Management Based on Qos." Digital Cont ent Technology and its Applications 4, no. 5 (August, 2010 2010): 55-61.
- [Moreno, 2010] Moreno, Jorge Jair , Liliam Paola Bolaños, and Manuel Alejandro Navia. "Exploración De Modelos Y Estándares De Calidad Para El Producto Software." UIS Ingenierías 9, no. 1 (Junio 2010): 39-53.
- [OASIS, 2010] OASIS. "Web Services Quality Factors Version 1.0." OASIS, <http://docs.oasis-open.org/wsrm/wsrf/v1.0/WS-Quality-Factors-v1.0-cd02.html>.
- [Oberortner, 2010] Oberortner, Ernst, Uwe Zdun, and Schahram Dustdar. "Patterns for Measuring Performance-Related Qos Properties in Distributed Systems." Paper presented at the 17th Conference on Pattern Languages of Programs (PLOP), Nevada, USA, 2010.
- [Rajendran, 2010] Rajendran, T, P Balasubramanie, and Resmi Cherian. "An Efficient

- Ws-Qos Broker Based Architecture for Web Services Selection." International Journal of Computer Applications IJCA 1, no. 9 (2010): 75-80.
- [W3C, 2010] W3C. "Guía Breve De Servicios Web." <http://www.w3c.es/Divulgacion/GuiasBreves/ServiciosWeb>.
- [Wang, 2010] Wang, Lizhe, Gregor VON LASZEWSKI, Marcel KUNZE, and Jie TAO. "Cloud Computing: A Perspective Study." New Generation Computing 28, no. 2 (2010): 137-46.
- [Wei, 2010] Wei, Yi, and M. Brian Blake. "Service-Oriented Computing and Cloud Computing: Challenges and Opportunities." IEEE Computer Society 14, no. 6 (Nov.-Dec. 2010): 72-75.
- [Zhang, 2010] Zhang, Qi, Lu Cheng, and Raouf Boutaba. "Cloud Computing: State-of-the-Art and Research Challenges." J Internet Serv Appl (2010) 1, no. 1 (2010-05-01 2010): 7-18.
- [Zhang, 2010] Zhang, Shuai, Shufen Zhang, Xuebin Chen, and Xiuzhen Huo. "The Comparison between Cloud Computing and Grid Computing ". International Conference on Computer Application and System Modeling (ICCASM 2010) 11 (2010): 72-75.
- [Zheng, 2010] Zheng, Zibin, Yilei Zhang, and Michael R Lyu. "Distributed Qos Evaluation for Real-World Web Services." Paper presented at the Web Services (ICWS), 2010 IEEE International Conference on, 2010.
- [Alrifai, 2009] Alrifai, Mohammad, and Thomas Risse. "Combining Global Optimization with Local Selection for Efficient Qos-Aware Service Composition." Track: Web Engineering / Session: Service Oriented Development (2009): 881-90.
- [Bejar, 2009] Bejar, Oscar Jair Cabrera. "Banco De Pruebas Orientado a La Calidad O Qos Para Apoyar La Selección Y Composición De Servicios Web ", cenidet, 2009.
- [Harauz, 2009] Harauz, John, Lori M. Kaufman, and Bruce Potter. "Data Security in the World of Cloud Computing." IEEE Security & Privacy 7, no. 4 (July-

- Aug 2009): 61-64.
- [Jensen, 2009] Jensen, Meiko, Jorg Schwenk, Nils Gruschka, and Luigi Lo Iacono. "On Technical Security Issues in Cloud Computing." Paper presented at the Cloud Computing, 2009. CLOUD'09. IEEE International Conference on, 2009.
- [Li, 2009] Li, Meng, Hao-peng Chen, and Nan Wang. "The Description and Calculation of Quality of Composite Services." In APSCC 2009, 385-90: IEEE Asia-Pacific, 2009.
- [Pearson, 2009] Pearson, Siani. "Taking Account of Privacy When Designing Cloud Computing Services." In ICSE Workshop on Software Engineering Challenges of Cloud Computing, 2009. CLOUD '09. , 44 - 52. Vancouver, BC: IEEE, 2009.
- [Rajendran, 2009] Rajendran, T, and P Balasubramanie. "Analysis on the Study of Qos-Aware Web Services Discovery." arXiv preprint arXiv:0912.3965 (2009).
- [Rosenberg, 2009] Rosenberg, Florian, Predrag Celikovic, Anton Michlmayr, Philipp Leitner, and Schahram Dustdar. "An End-to-End Approach for Qos-Aware Service Composition." IEEE International Enterprise Distributed Object Computing Conference, 2009. EDOC '09. (1-4 Sept 2009): 151-60.
- [Vaquero, 2009] Vaquero, Luis M., Luis Roderio-Merino, Juan Caceres, and Maik Lindner. "A Break in the Clouds: Towards a Cloud Definition." ACM SIGCOMM Computer Communication Review 39 (2009): 50-55.
- [Xiong, 2009] Xiong, Kaiqi, and Harry Perros. "Service Performance and Analysis in Cloud Computing." In 2009 Congress on Services - I, 693-700: IEEE computer society, 2009.
- [Al-Masri, 2008] Al-Masri, Eyhab, and Qusay H Mahmoud. "Investigating Web Services on the World Wide Web." Paper presented at the Proceedings of the 17th international conference on World Wide Web, 2008.
- [Badr, 2008] Badr, Youakim, Ajith Abraham, Frédérique Biennier, and Crina Grosan. "Enhancing Web Service Selection by User Preferences of Non-

- Functional Features." Paper presented at the Next Generation Web Services Practices, 2008. NWESP'08. 4th International Conference on, 2008.
- [Buyya, 2008] Buyya, Rajkumar, Chee Shin Yeo, Srikumar Venugopal, James Broberg, and Ivona Brandic. "Cloud Computing and Emerging It Platforms: Vision, Hype, and Reality for Delivering Computing as the 5th Utility." *Future Generation Computer Systems* 25 (2009) (2008): 599–616.
- [Chaari, 2008] Chaari, Sodki, Youakim Badr, Frédérique Biennier, Chokri BenAmar, and Joël Favrel. "Framework for Web Service Selection Based on Non-Functional Properties." *International Journal of Web Services Practices* 3, no. 2 (2008): 94-109.
- [Diamadopoulou, 2008] Diamadopoulou, Vassiliki, Christos Makris, Yannis Panagis, and Evangelos Sakkopoulos. "Techniques to Support Web Service Selection and Consumption with Qos Characteristics." *Journal of Network and Computer Applications* 31, no. 2 (2008): 108-30.
- [Foster, 2008] Foster, Ian, Yong Zhao, Ioan Raicu, and Shiyong Lu. "Cloud Computing and Grid Computing 360-Degree Compared." In *Grid Computing Environments Workshop*, 2008. GCE '08, 1- 10 Austin, TX, 2008.
- [Lin, 2008] Lin, Wei-Li, Chi-Chun Lo, Kuo-Ming Chao, and Muhammad Younas. "Consumer-Centric Qos-Aware Selection of Web Services." *Journal of Computer and System Sciences* 74, no. 2 (2008): 211-31.
- [Mounla, 2008] Mounla, Rami. "Qos-Aware Web Service Composition." In *Computer Science Research Student*, 115-22. Christchurch, New Zealand: NZCSRSC, 2008.
- [Nurhayati, 2008] Nurhayati, Wan, WAN AB. RAHMAN, and Farid MEZIANE. "Challenges to Describe Qos Requirements for Web Services Quality Prediction to Support Web Services Interoperability in Electronic Commerce ". *Communications of the IBIMA* 4, no. 6 (2008): 50-58.
- [W3C, 2008] W3C. "El W3c De La a a La Z." <http://www.w3c.es/Divulgacion/a-z/>.

- [Yu, 2008] Yu, Hong Qing, and Stephan Reiff-Marganiec. "A Method for Automated Web Service Selection." In IEEE Congress on Services, 513- 20 Honolulu, HI: IEEE CONFERENCE PUBLICATIONS, 2008.
- [Ardagna, 2007] Ardagna, Danilo, and Barbara Pernici. "Adaptive Service Composition in Flexible Processes." IEEE TRANSACTIONS ON SOFTWARE ENGINEERING 33, no. 6 (JUNE 2007): 369-84.
- [Buschmann, 2007] Buschmann, F., K. Henney, and D.C. Schmidt. Pattern-Oriented Software Architecture: A Pattern Language for Distributed Computing, Volume 4. Wiley India Pvt. Limited, 2007.
- [De Cock, 2007] De Cock, Martine, Sam Chung, and Omar Hafeez. "Selection of Web Services with Imprecise Qos Constraints." Paper presented at the Web Intelligence, IEEE/WIC/ACM International Conference on, 2007.
- [Hwang, 2007] Hwang, San-Yih, Haojun Wang, Jian Tang, and Jaideep Srivastava. "A Probabilistic Approach to Modeling and Estimating the Qos of Web-Services-Based Workflows." Information Sciences 177, no. 23 (1 December 2007 2007): 5484–503.
- [Jurca, 2007] Jurca, Radu, Boi Faltings, and Walter Binder. "Reliable Qos Monitoring Based on Client Feedback." Paper presented at the Proceedings of the 16th international conference on World Wide Web, 2007.
- [Moreno, 2007] Moreno, Jorge Jair , Hugo Hernando Andrade Sosa, and Liliam Bolaños. "Compilación De Un Modelo Para Evaluar Atributos De Calidad En Productos Software ". Enlace Informático 6, no. 1 (Diciembre 2007): 1-14.
- [Tian, 2007] Tian, M. , A. Gramm, H. Ritter, J. Schiller, and R. Winter "A Survey of Current Approaches Towards Specification and Management of Quality of Service for Web Services." De Gruyter, December 2007 2007, 132–39.
- [Wang, 2007] Wang, Hongbing, Ping Tong, and Phil Thompson. "Qos-Based Web Services Selection " In ICEBE 2007, 631-37: IEEE Computer Society 2007.

- [Wang, 2007] Wang, Yao, and Julita Vassileva. "A Review on Trust and Reputation for Web Service Selection." In 27th International Conference on Distributed Computing Systems Workshops (ICDCSW'07), 25. Toronto, Ont.: IEEE Computer Society, 2007.
- [Yu, 2007] Yu, TAO, YUE Zhang, and Kwe i-Jay Lin. "Efficient Algorithms for Web Services Selection with End-to-End Qos Constraints." ACM Transactions on the Web 1, no. 1 (May 2007): 1-26.
- [Yu, 2007] Yu, Weider D., Rachana B. Radhakrishna, Sumana Pingali, and Vijaya Kolluri. "Modeling the Measurements of Qos Requirements in Web Service Systems." SIMULATION 83, no. 1 (January 2007 2007): 75–91.
- [Zeng, 2007] Zeng, Liangzhao, Hui Lei, and Henry Chang. "Monitoring the Qos for Web Services ", 132-44: Springer Berlin / Heidelberg, 2007.
- [Marchese, 2007] Marchese, Mario. Qos over Heterogeneous Networks. Wiley, 2007.
- [Chen, 2006] Chen, Yan-ping, Zeng-zhi Li, Qin-xue Jin, and Chuang Wang. "Study on Qos Driven Web Services Composition ". APWeb 2006 3841 (2006): 702–07.
- [Garcia, 2006] Garcia, Diego Zuquim Guimaraes, and Maria Beatriz Felgar de Toledo. "A Web Service Architecture Providing Qos Management." Paper presented at the Web Congress, 2006. LA-Web'06. Fourth Latin American, 2006.
- [Jaeger, 2006] Jaeger, Michael C, and Hendrik Ladner. "A Model for the Aggregation of Qos in Ws Compositions Involving Redundant Services." Journal of Digital Information Management 4, no. 1 (2006): 44.
- [Jang, 2006] Jang, J.H., D.H. Shin, and K.H. Lee. "Fast Quality Driven Selection of Composite Web Services." (2006).
- [Rosenberg, 2006] Rosenberg, Florian, Christian Platzer, and Schahram Dustdar. "Bootstrapping Performance and Dependability Attributes Ofweb Services." Paper presented at the Web Services, 2006. ICWS'06. International Conference on, 2006.

- [Vu, 2006] Vu, Le-Hung, Manfred Hauswirth, Fabio Porto, and Karl Aberer. "A Search Engine for Qos-Enabled Discovery of Semantic Web Services." *International Journal of Business Process Integration and Management* 1, no. 4 (2006): 244-55.
- [Agarwal, 2005] Agarwal, Vikas, Koustuv Dasgupta, Neeran Karnik, and Arun Kumar. "A Service Creation Environment Based on End to End Composition of Web Services." In *WWW '05 Proceedings of the 14th international conference on World Wide Web*, 128-37 NY, USA: ACM New York, USA, 2005.
- [Canfora, 2005] Canfora, Gerardo, Massimiliano Di Penta, Raffaele Esposito, and Maria Luisa Villani. "An Approach for Qos-Aware Service Composition Based on Genetic Algorithms." In *GECCO'05*, 1069-75. Washington, DC, USA: ACM, 2005.
- [Frankova, 2005] Frankova, Ganna. "Web Service Quality Composition Modelling." *Citeseer*, 2005, 67-72.
- [Jaeger, 2005] Jaeger, Michael C., Gregor Rojec-Goldmann, and Gero Mühl. "Qos Aggregation in Web Service Compositions." In *e-Technology, e-Commerce and e-Service (EEE '05)*, 181-85: IEEE, 2005.
- [O'Brien, 2005] O'Brien, Liam, Len Bass, and Paulo F. Merson. "Quality Attributes and Service-Oriented Architectures." *Software Engineering Institute* 2005.
- [OASIS, 2005] OASIS. "Quality Model for Web Services." 1-43: OASIS, 2005.
- [Taher, 2005] Taher, L, H El Khatib, and R Basha. "A Framework and Qos Matchmaking Algorithm for Dynamic Web Services Selection." Paper presented at the *Second International Conference on Innovations in Information Technology (IIT'05)*, Dubai, UAE, 2005.
- [Thio, 2005] Thio, Niko, and Shanika Karunasekera. "Automatic Measurement of a Qos Metric for Web Service Recommendation " In *2005 Australian Software Engineering Conference*, 202-11 IEEE Computer Society, 2005.
- [Völter, 2005] Völter, M., M. Kircher, and U. Zdun. *Remoting Patterns: Foundations of*

- Enterprise, Internet and Realtime Distributed Object Middleware. Wiley, 2005.
- [Yu, 2005] Yu, Tao, and Kwe i-Jay Lin. "A Broker-Based Framework for Qos-Aware Web Service Composition " In e-Technology, e-Commerce and e-Service, 22- 29 IEEE, 2005.
- [Yu, 2005] Yu, Tao, and Kwe i-Jay Lin. "Service Selection Algorithms for Web Services with End-to-End Qos Constraints." INFORMATION SYSTEMS AND E-BUSINESS MANAGEMENT 3, no. 2 (2005): 103-26.
- [Cardoso, 2004] Cardoso, Jorge, Amit Sheth, John Miller, Jonathan Arnold, and Krys Kochut. "Quality of Service for Workflows and Web Service Processes." Web Semantics: Science, Services and Agents on the World Wide Web 1, no. 3 (2004): 281-308.
- [Gamma, 2004] Gamma, E. Design Patterns: Elements of Reusable Object-Oriented Software. Pearson Education, 2004.
- [Garofalakis, 2004] Garofalakis, John, Yannis Panagis, Evangelos Sakkopoulos, and Athanasios Tsakalidis. "Web Service Discovery Mechanisms: Looking for a Needle in a Haystack?" Paper presented at the International Workshop on Web Engineering, 2004.
- [Jaeger, 2004] Jaeger, Michael C., Gregor Rojec-Goldmann, and Gero Muehl. "Qos Aggregation for Web Service Composition Using Workflow Patterns." In Enterprise Distributed Object Computing Conference, 149-59 Eighth IEEE International, 2004.
- [Liu, 2004] Liu, Yutu, Anne H.H. Ngu, and Liangzhao Zeng. "Qos Computation and Policing in Dynamic Web Service Selection." In WWW2004, 66-73: ACM New York, NY, USA, 2004.
- [Maximilien, 2004] Maximilien, E. Michael, and Munindar P. Singh. "A Framework and Ontology for Dynamic Web Services Selection." IEEE INTERNET COMPUTING, SEPTEMBER - OCTOBER 2004 2004, 84-93.
- [Menascé, 2004] Menascé, Daniel A. "Qos Issues in Web Services." IEEE INTERNET

- COMPUTING 6, no. 6 (Nov/Dec 2002): 72-75.
- [Tian, 2004] Tian, M., A. Gramm, H. Ritter, and J. Schiller. "Efficient Selection and Monitoring of Qos-Aware Web Services with the Ws-Qos Framework." In *Web Intelligence, 2004*, 152-58 IEEE/WIC/ACM International Conference on, 2004.
- [W3C, 2004] W3C. "Web Services Glossary." <http://www.w3.org/TR/ws-gloss/>.
- [Zeng, 2004] Zeng, Liangzhao, Boualem Benatallah, and Anne H.H. Ngu. "Qos-Aware Middleware for Web Services Composition." *IEEE Transactions on Software Engineering* 30, no. 5 (MAY 2004): 311-27.
- [Kalepu, 2003] Kalepu, Sravanthi, Shonali Krishnaswamy, and Seng Wai Loke. "Verity: A Qos Metric for Selecting Web Services and Providers " In *Fourth International Conference on Web Information Systems Engineering Workshops*, 131-39 IEEE Computer Society, 2003.
- [Keller, 2003] Keller, Alexander, and Heiko Ludwig. "The Wsla Framework: Specifying and Monitoring Service Level Agreements for Web Services." *Journal of Network and Systems Management* 11, no. 1 (2003): 57-81.
- [Papazoglou, 2003] Papazoglou, M. P., and D. Georgakopoulos. "Service-Oriented Computing." *COMMUNICATIONS OF THE ACM* 46 (2003): 25-28.
- [Patel, 2003] Patel, Chintan, Kaustubh Supekar, and Yugyung Lee. "A Qos Oriented Framework for Adaptive Management of Web Service Based Workflows." *Database and Expert Systems Applications* 2736 (2003): 826-35.
- [Peltz, 2003] Peltz, Chris. "Web Services Orchestration and Choreography." *IEEE Computer Society*, Oct. 2003 2003, 46-52
- [Ran, 2003] Ran, Shuping. "A Model for Web Services Discovery with Qos." *ACM SIGecom Exchanges* 4, no. 1 (Spring, 2003 2003): 1-10.
- [W3C, 2003] W3C. "Qos for Web Services: Requirements and Possible Approaches." 2003.

-
-
- [Yang, 2003] Yang, Jian. "Webservice Componentization." ACM, October 2003 2003, 35-40.
- [Zeng, 2003] Zeng, Liangzhao, Boualem Benatallah, Marlon Dumas, Jayant Kalagnanam, and Quan Z. Sheng. "Quality Driven Web Services Composition." In 12th International Conference on the World Wide Web (WWW). Budapest, Hungary: ACM Press, 2003.
- [Jin, 2002] Jin, Li-jie, Vijay Machiraju, and Akhil Sahai. "Analysis on Service Level Agreement of Web Services." HP June (2002).
- [Leymann, 2002] Leymann, F., D. Roller, and M.-T Schmidt. "Web Services and Business Process Management." IBM SYSTEMS JOURNAL 41 (2002): 198-211.
- [Menascé, 2002] Menascé, Daniel A. "Composing Web Services: A Qos View." IEEE INTERNET COMPUTING 8, no. 6 (NOVEMBER - DECEMBER 2004): 80- 90
- [Narayanan, 2002] Narayanan, Srini, and Sheila A. McIlraith. "Simulation, Verification and Automated Composition of Web Services " ACM, May 7-11, 2002 2002, 77-88.
- [Newcomer, 2002] Newcomer, Eric. Understanding Web Services Xml, Wsdl, Soap, and Uddi. Indianapolis: Pearson Education Corporate Sales 2002.
- [Pressman, 2002] Pressman, Roger S. Ingenieria De Software Un Enfoque Práctico. Madrid: McGraw-Hill, 2002.
- [Sahai, 2002] Sahai, Akhil, Anna Durante, and Vijay Machiraju. "Towards Automated Sla Management for Web Services." Hewlett-Packard Research Report HPL-2001-310 (R. 1) (2002).
- [Sahai, 2002] Sahai, Akhil, Vijay Machiraju, Mehmet Sayal, Aad Van Moorsel, and Fabio Casati. "Automated Sla Monitoring for Web Services." Management Technologies for E-Commerce and E-Business Applications (2002): 28-41.
- [Yang, 2002] Yang, Jian, and Mike. P. Papazoglou. "Web Component: A Substrate

- for Web Service Reuse and Composition." 21-36. Berlin Heidelberg: Springer Berlin / Heidelberg, 2002.
- [Schmidt, 2000] Schmidt, D.C., M. Stal, H. Rohnert, and F. Buschmann. Pattern-Oriented Software Architecture, Patterns for Concurrent and Networked Objects. John Wiley & Sons, 2000.
- [Buschmann, 1999] Buschmann, F. Pattern Oriented Software Architecture: A System of Patters. John Wiley&Sons, 1999.
- [ISO8402, 1994] ISO8402. "Gestión De La Calidad Y Aseguramiento De La Calidad." 1-30: UNE-EN, 1994.
- [Ejiogu, 1991] Ejiogu, Lem O. Software Engineering with Formal Metrics ilustrada ed. 1991.
- [IEEE, 1990] IEEE. "Ieee Standard Glossary of Software Engineering Terminology." IEEE, 1990.