

## **Caracterización del manejo de mensajes para la orquestación de servicios web**

**Evaristo Arroyo F.\*; Rosario Ch. Peley\*\* y Jorge Finol\*\*\***

### **Resumen**

Hoy en día las organizaciones requieren de habilidad para adaptarse a los requerimientos de los clientes y a las condiciones cambiantes del entorno. Para ello se utilizan todas las herramientas y técnicas disponibles. Con este propósito surgen los Servicios Web los cuales constituyen un paradigma emergente para el diseño de sistemas software distribuido. Este trabajo se enfoca en el estudio del modelo orientado al mensaje, en el cual ocurre el “direccionamiento de mensajes”, ya que esto constituye un elemento fundamental en la orquestación de los Servicios Web. Igualmente, se discutirá sobre algunas características de la implementación de las tecnologías utilizadas en el manejo de los mensajes, considerando las posibles debilidades y potencialidades de este modelo. La investigación es descriptiva, utilizando el análisis documental para la obtención de resultados. En el estudio se demuestra que el modelo para el manejo de mensajes presenta características que pudieran dificultar la composición de los mismos.

**Palabras clave:** Servicio Web, Modelo orientado a Mensajes, SOA, SOAP, HTTP.

### *Characterization of Message Handling for the Orchestration of Web Services*

### **Abstract**

Nowadays, organizations need the ability to adapt themselves to the requirements of clients and the changing conditions of the environment. To do this, they use all tools and techniques available. Web Services have arisen

\* Profesor del Departamento de Investigación e Información de la Facultad de humanidades y Educación. Universidad del Zulia. Correo electrónico: evarroyo@hdes.luz.edu.ve

\*\* Lic. En Educación Mención Ciencias Pedagógicas. Área Tecnología Instrumental. Mag. En Planificación educativa. Universidad del Zulia. Correo electrónico: rospeley@gmail.com

\*\*\* Ingeniero en Informática. Universidad del Zulia. Correo electrónico: jorgefinol@gmail.com

for this purpose; they constitute an emerging paradigm for the design of distributed software systems. This paper focuses on studying the model orientated to the message where the “routing of messages” occurs, because it constitutes a fundamental element in the orchestration of Web Services. Likewise, it will discuss some characteristics of implementing the technologies used in message handling, considering the possible weaknesses and potentials of the model. The investigation is descriptive, using documentary analysis for obtaining results. The study demonstrates that the message handling model presents characteristics that could make message composition difficult.

**Key words:** Web services, message oriented model, SOA, SOAP, HTTP.

## **1. Introducción**

El desarrollo de la Web ha continuado, y con ella las nociones de distribución, colaboración y flexibilidad han dado origen a ideas que permiten un mejor y mayor uso de los recursos de la red. Es así que surge la Arquitectura Orientada a Servicios (Service Oriented Architecture, SOA) la cual es una arquitectura de software que define la utilización de servicios como construcciones básicas para el desarrollo de aplicaciones.

Estos Servicios Web, base fundamental de la arquitectura, encapsulan tanto la lógica de negocios como los datos relevantes a la solución para la cual fueron diseñados. Sin embargo, no son sistemas autosuficientes y cerrados, sino por el contrario, su principio es la colaboración e intercambio (composición) con otros Servicios Web, los cuales pueden pertenecer a diferentes propietarios.

Esta característica añade versatilidad y flexibilidad a los sistemas distribuidos, pero también crea un problema de interoperabilidad de los servicios, es decir, de compatibilidad de los sistemas. Este es un factor crítico que debe ser resuelto para lograr la evolución de esta tecnología así como también la adopción de la misma.

En virtud de lo anterior, se hace necesario estudiar la manera de lograr la colaboración exitosa de los servicios Web, de tal forma que su composición permita un uso útil de esta tecnología.

## **2. Los servicios Web**

Los servicios Web son definidos como aplicaciones de software que pueden ser descubiertas, descritas y accedidas a través de intranets, extranets e Internet, utilizando XML y protocolos Web estándares (Dacosta, Obrst y Smith, 2003). Estos servicios, cuyas capas pueden verse en la Figura 1, tienen una interfaz descrita en un formato procesable por los ordenadores (específicamente el Web Service Description Language, WSDL), lo cual caracteriza su beneficio, es decir, la posibilidad de encapsular su funcionalidad dentro de una interfaz pública.

**Figura 1**  
**Capaz básicas de los servicios Web**



**Fuente:** (Daconta et al. 2003).

Otros sistemas pueden interactuar con estos de la manera indicada por su descripción, utilizando para ello mensajes SOAP (Simple Object Access Protocol), con HTTP para la transmisión de los mensajes y el formato XML como formato de representación (Booth, Haas, McCabe, Newcomer, Champion, Ferris y Orchard, 2004).

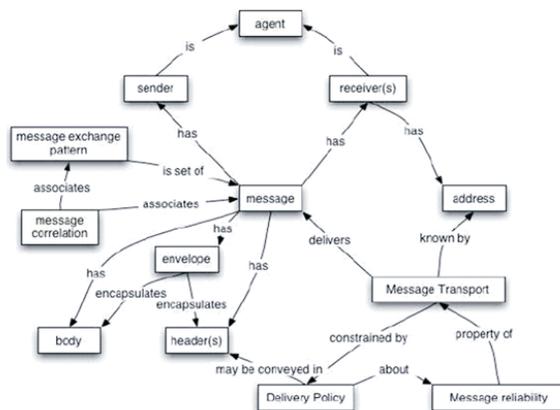
La arquitectura básica de los Servicios Web incluye tecnologías capaces de: intercambiar mensajes, describir servicios Web, publicar y descubrir la descripción de estos servicios. Esta arquitectura modela la interacción entre los roles de: proveedor de servicio, descriptor y publicador del servicio, y solicitante del servicio.

Un Servicio Web es un concepto abstracto que es implementado mediante agentes software. Mientras el servicio es el recurso abstracto caracterizado por una serie de funcionalidades, los agentes son los elementos concretos (reales) que lo implementa mediante el intercambio de mensajes. Por lo tanto, el primer desafío de este modelo es la necesidad de acuerdos en la semántica (significado) y en los mecanismos para el intercambio de los mensajes.

### 3. Características del modelo orientado al mensaje

En el modelo orientado a mensajes de la arquitectura de los Servicios Web (Figura 2), el mensaje tiene un "sobre o cubierta" que encapsula al cuerpo del mensaje, y posiblemente, a una o más cabeceras. Esta cubierta sirve para suministrar al servicio de transporte la información de direccionamiento necesaria. La cabecera puede incluir información pertinente para ampliar la funcionalidad del servicio Web, tal como seguridad, contexto de transacción, información de orquestación o de enrutamiento de mensajes (Champion, Ferris, Newcomer y Orchard 2002). El cuerpo contiene el mensaje propiamente o URIs a los recursos de la data.

**Figura 2**  
**Modelo orientado al mensaje**



**Fuente:** (Booth et al. 2004).

Hay que notar que la responsabilidad del envío y entrega del mensaje es del mecanismo de transporte de mensaje. Se observa, además, que no existe ninguna relación, al menos explícita, entre esta cubierta y la dirección del mensaje, no hay anotación en el mensaje sobre la dirección del receptor.

Aun más, en la documentación oficial de la arquitectura (Booth et al. 2004), se indica que la cubierta del mensaje “puede” contener la dirección del receptor del mensaje, lo cual no es un requerimiento. Esta característica del modelo crea relaciones débiles, no formales, entre la dirección y el mensaje, originando probables fallas donde el estándar no daría respuesta o faltaría por definir, posibilitando el dar soluciones diversas no ajustadas al canon del estándar y dificultando, de esta manera, el proceso de correlación de los mensajes, y por lo tanto, la composición u orquestación de los servicios Web.

### 3.1. Asociación del mensaje con la dirección de destino

Un mensaje, de alguna manera, debe ser asociado con la dirección del receptor del mismo. Esta combinación, mensaje-dirección\_destino, parece ser parte importante de la arquitectura, no obstante, ni aun SOAP requiere que la dirección sea incluida en la cabecera del mensaje, ya que esto puede ser manejado por el mecanismo de transporte utilizado.

La cubierta del mensaje puede contener información necesaria para efectuar la entrega del mensaje, al menos, debe contener suficiente información de direccionamiento para que el mecanismo de transporte pueda entregar el mensaje. Comúnmente esta información es parte de la data incrustada en un documento WSDL, lo cual no lo convierte en un estándar. Además, en la “cubierta” del mensaje también puede incluirse información de seguridad que permita al mensaje ser autenticado y anota-

do sobre la calidad de servicio. Un mecanismo de transporte bien diseñado debe ser capaz de entregar un mensaje basándose para ello sólo en la información del sobre.

De esta manera, un mensaje encriptado, que proteja la identidad del remitente, del receptor y del propio contenido del mensaje, debe ser posible entregarlo utilizando para ello solamente la información de direccionamiento disponible en la cubierta.

Como se indica, los mensajes propiamente no poseen información para su direccionamiento. Para el envío de estos se utiliza el concepto de "transporte de mensaje" (generalmente HTTP). Este elemento debe conocer la información de direccionamiento para lograr la entrega del mensaje. Pero además, en el proceso de resolución y manejo de los mensajes es necesario que el Servicio Web registre información que le permita responder y correlacionar estos mensajes, lo cual constituye un requisito para lograr la composición, y orquestación, de servicios.

Esta situación puede perjudicar el funcionamiento del servicio, así como su composición con otros, ya que, de existir ambigüedad en la información de quien se recibe o a quien se envía el mensaje, puede afectar la correlación de estos, haciendo que la orquestación sea una tarea inútil e imposible. Más aun, la arquitectura de los servicios Web no incluyen el concepto de correlación automática de solicitudes y respuestas, la cual es definida comúnmente por la aplicación, como si lo hacen algunas tecnologías RPC (Remote Procedure Call). La correlación de los mensajes de solicitud y respuesta es comúnmente definida por la aplicación (Champion et al. 2002).

Cuando la correlación debe ser manejada de forma explícita, se recurre a la técnica de asociar un identificador al propio mensaje. Este identificador es un registro que permite correlacionar el mensaje recibido con la petición original. El remitente puede adicionar también un identificador para un servicio, quien será el receptor del mensaje.

Además, la correlación puede realizarse mediante el protocolo subyacente; por ejemplo HTTP/1.1 permite correlacionar la solicitud con su respuesta, sin embargo, este no dispone de mecanismos para asegurar la entrega o capacidad asíncrona, lo cual constituye otro aspecto a considerar en el logro de la orquestación de Servicios Web. En caso de requerirse esta característica, entonces se debe recurrir a soluciones propietarias como Websphere MQ de IBM.

### **3.2. Servicios Web síncronos y asíncronos**

Los Servicios Web pueden ser de tipo síncrono o asíncrono. Los procesos son síncronos cuando a cada solicitud corresponde una respuesta inmediata, de esta manera, es posible saber de forma instantánea si una solicitud tuvo, o no, éxito. Sin embargo, en la realidad Web, la ejecución de un proceso puede llevar al intercambio entre varios servicios diferentes; de tal manera que el intercambio síncrono no es suficiente en un ambiente donde un proceso puede tener que esperar la respuesta (debido a

falla, a disponibilidad, o cualquier otro factor que afecte al servicio proveedor) ya que cualquier factor de retardo puede hacer que el proceso se detenga, evitando así que el servicio se cumpla.

Al momento de considerar la composición de servicios Web, es necesario tener en cuenta la posibilidad de que estos intercambien mensajes de forma asíncrona. Es por esto que aunque inicialmente SOAP se basaba en transmisiones síncronas, en la actualidad esta tendencia se mueve en otra dirección en virtud de la necesidad de permitir la composición de servicios. En el intercambio de mensajes asíncronos cada operación de comunicación es una unidad de trabajo autocontenida e independiente. De esta manera, el proceso de arranque sólo se interesa en la iniciación de la conversación, en el entendido de que eventualmente recibirá una respuesta, de forma asíncrona.

Este tipo de procesos permite, y requiere, un sistema centralizado de administración de fallas. Además, los servicios asíncronos, traen consigo el concepto de calidad de servicio al permitir asegurar la entrega del mensaje.

Además de la sincronía, o no, de los mensajes, es posible distinguir diferentes tipos de interacciones soportadas, como:

1. Una vía, en este caso el solicitante envía un mensaje al proveedor el cual puede o no retornar una respuesta. En caso de hacerlo, el solicitante puede que no la escuche o cierre el canal de comunicación, en todo caso la respuesta no será procesada por el solicitante.
2. Conversacional, el servicio solicitante y proveedor intercambian múltiples mensajes. Esto es definido por lenguajes de coreografía.
3. Mucho a muchos, el solicitante envía mensajes a muchos proveedores, o el servicio proveedor responde a muchos solicitantes. También se logra mediante lenguajes de coreografía.

Todos estos factores deben ser tomados en cuenta para lograr un acertada composición de los servicios Web.

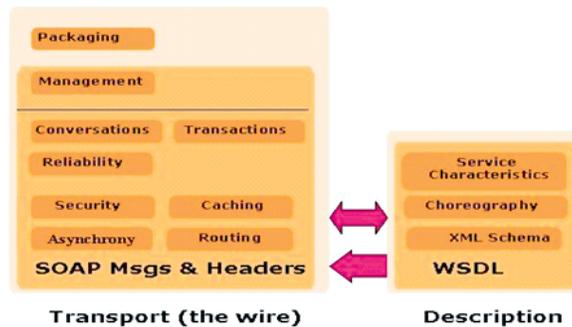
#### **4. Implementación del modelo**

En la práctica, para lograr la implementación del modelo, y considerando sus características, se recurre a la aplicación de los siguientes principios y tecnologías (Figura 3).

1. Enrutamiento de mensajes mediante especificaciones dinámicas de rutas de mensajes entre nodos.
2. Correlación. Para lograr esto se pueden utilizar dos estrategias, la obvia mediante HTTP o mediante ID SOAP module.

Centrando la atención en el bloque denominado “wire” de la Figura 3, se observa que esta sección encapsula los conceptos y tecnologías que se encargan del intercambio de información entre los roles indicados al principio. Esta pila incluye lo relacionado con el transporte de red, el empaquetamiento de mensajes y la extensión de los mismos, que pueden ser utilizadas para facilitar el intercambio de datos. Este

**Figura 3**  
**Aplicación de tecnologías en la implementación de la arquitectura**



**Fuente:** (Champion et al. 2002).

bloque involucra a tres elementos fundamentales como son el transporte, el empaquetamiento y las extensiones, todos estos se discuten a continuación.

#### 4.1. Transporte

El fundamento de los servicios Web es la red. Los servicios Web deben ser accesibles e invocables por otro a través de la red. Para ello, y debido a su ubicuidad, HTTP es el protocolo de red estándar para servicios Web disponibles en Internet, aunque no es el único, ya que pueden ser soportados el SMTP y FTP.

La elección concreta del protocolo de red depende por completo de los requerimientos de aplicación, los cuales incluyen aspectos tales como la seguridad, disponibilidad, rendimiento y fiabilidad. Esto permite que los servicios web existentes para sacar mayor provecho de las infraestructuras de redes y del middleware orientado a mensajes, como MQSeries.

Es por ello que dentro de una empresa con múltiples tipos de infraestructura de red, HTTP puede ser usado como un común e interoperable puente para conectar los distintos sistemas. Uno de los beneficios de los servicios Web es que estos proveen un modelo de programación único para el desarrollo y uso tanto de Intranet privadas, como de servicios de Internet públicos. Como resultado, la selección de la tecnología de red puede ser completamente transparente para el desarrollador y consumidor del servicio.

#### 4.2. Empaquetamiento

Representa las tecnologías que pueden utilizarse para empaquetar la información que será intercambiada. XML ha sido adoptada ampliamente como base para protocolos de empaquetamiento de mensajes de servicios Web. En este sentido, SOAP es un sencillo y liviano mecanismo, basado en XML, para la creación de paquetes de datos estructurados que

pueden ser intercambiados entre aplicaciones de red. SOAP consiste de cuatro componentes fundamentales: una cubierta que define un marco para describir la estructura de mensajes; un conjunto de reglas de codificación para expresar instancias de aplicaciones; una convención para representar llamadas a procedimientos remotos (RPC) y respuestas; y un conjunto de reglas para usar SOAP con HTTP. SOAP puede ser usada con una gran variedad de protocolos de red, tales como, HTTP, FTP, RMI/IIOP, o propietarios protocolos de mensajes. SOAP es el estándar actual para mensajes XML en virtud de varias razones:

1. Es relativamente simple, definiendo una capa que se construye sobre tecnologías de red existentes, tal como HTTP, la cual es ampliamente aplicada.
2. Es flexible y extensible, en el sentido de que trata de resolver los problemas afrontados por los desarrolladores cuando construyen servicios web; además, provee un marco de trabajo que permite soluciones incrementales en la medida de la necesidad.
3. SOAP está basado en XML y disfruta de un amplio soporte de la industria y la comunidad de desarrolladores.

#### **4.3. Extensión**

Esta, construida sobre las anteriores capas, provee un marco que permite que información adicional sea unida a los mensajes del servicio Web representando una variedad de propósitos adicionales, tales como, contexto, enrutamiento, políticas, etc. Como parte clave de la estructura de la cubierta del mensaje, SOAP define un mecanismo para incorporar extensiones ortogonales (también conocidas como características) a los mensajes en la forma de encabezados y reglas codificadas (algunas son: mensajes asíncronos, archivos adjuntos, patrón de intercambio de mensajes, correlación, autenticación, confidencialidad, integridad, enrutamiento, etc). Se espera que, como los servicios Web son adoptados y evolucionan, una amplia colección de tales extensiones emergerán y serán estandarizadas.

## **5. Conclusiones**

Los servicios Web constituyen una tecnología emergente de suma utilidad en el ambiente empresarial moderno, mediante el cual se puede compartir con los socios de negocio parte de los procesos relevantes del mismo, todo mediante tecnologías estándares de Internet, sin necesidad de exponer información estratégica. En general permiten acceder a soluciones y aplicaciones de forma transparente al usuario.

Este modelo de aplicación se basa en el envío y recepción de mensajes, por lo tanto se requiere que los servicios Web dispongan de la capacidad para el manejo de los mismos, de forma asíncrona y segura, lo cual permitirá la correlación y composición de los mismos, permitiendo así aumentar la funcionalidad de estos servicios.

El modelo diseñado para el manejo de mensajes presenta características que pudieran dificultar la composición de los mismos, sin embargo, con las tecnologías y protocolos de red es posible solucionar cualquier deficiencia que la arquitectura presenta.

### **Referencias Bibliográficas**

- BERNERS-LEE, Timothy; FIELDING, Roy y MASINTER, Larry (1998). Uniform Resource Identifiers (URI): Generic Syntax RFC Editor. <http://www.rfc-editor.org/rfc/rfc2396.txt>. Extraído el 03 de junio de 2008.
- BOOTH, David; HAAS, Hugo; MCCABE, Francis; NEWCOMER, Eric, CHAMPION, Michael; FERRIS, Chris y ORCHARD, David (2004). Web Services Architecture. <http://www.w3.org/TR/ws-arch>. Extraído el 25 de junio de 2008.
- CHAMPION, Michael; FERRIS, Chris; NEWCOMER, Eric y ORCHARD, David (2002). Web Services Architecture. <http://www.w3.org/TR/2002/WD-ws-arch-20021114/>, (2002). Extraído el 1 de julio de 2008.
- DACONTA, Michael; OBRST, Leo y SMITH, Kevin (2003). **The Semantic Web: A Guide to the Future of XML, Web Services, and Knowledge Management**. Indiana, United State. *Wiley Publishing, inc.*
- SHEN, Jun; GROSSMANN, Georg; YANG, Yun; STUMPTNER, Markus; SCHREFL, Michael y REITER, Thomas (2007). "Analysis of business process integration in web service context". **Future Generation Computer Systems**, Vol. 23, No. 3, Australia. Universidad de Wollongong, pp. 283-294.
- YEFIM, Natis (2003). Service-Oriented Architecture Scenario. <http://www.gartner.com/resources/114300/114358/114358.pdf>. Extraído el 15 de junio de 2006.