

MIDDIS: ARQUITECTURA DE REFERENCIA PARA LA INTERACCIÓN DE SERVICIOS BASADOS EN SOA E IMS

MIDDIS: A REFERENCE ARCHITECTURE FOR SOA-AND-IMS BASED SERVICES INTERACTION

Ximena Velasco Melo

Ing. en Electrónica y Telecomunicaciones, Estudiante de Maestría, Facultad de Ingeniería Electrónica y Telecomunicaciones, Universidad del Cauca, Grupo de Ingeniería Telemática W@PColombia, Cauca, Colombia. xvelasco@unicauca.edu.co

Óscar Mauricio Caicedo Rendón

Ing. en Electrónica y Telecomunicaciones, Especialista en Redes y Servicios Telemáticos, Magíster en Ingeniería, Área Telemática, Docente del Departamento de Telemática, Facultad de Ingeniería Electrónica y Telecomunicaciones, Universidad del Cauca, Grupo de Ingeniería Telemática W@PColombia, Cauca, Colombia. omcaicedo@unicauca.edu.co

Fecha de recepción: 15 de octubre de 2010

Fecha de aprobación: 9 de diciembre de 2010

RESUMEN

En telecomunicaciones la tendencia actual está dirigida hacia una búsqueda de la convergencia de redes fijas y móviles, y por lo tanto, las redes que se diseñan son más complejas. Así mismo, se presentan nuevos retos en el campo de la interconexión e integración de servicios a través de múltiples redes, tecnologías y áreas de negocio, lo cual hace imprescindible interoperar los servicios de las Tecnologías de Información (Information Technologies, IT), con los de telecomunicaciones. Para aportar en la solución de estos retos y debido además, a la ausencia de un entorno de telecomunicaciones convergente y completamente adecuado para la prestación de servicios tradicionales y nuevos, en este artículo se presenta una arquitectura de referencia que permite la habilitación y entrega rápida de servicios convergentes para el mundo IT y el mundo de las Telecomunicaciones, con la mediación en la interacción de servicios basados en la Arquitectura Orientada a Servicios (Service Oriented Architecture, SOA), y el Subsistema Multimedia IP (IP Multimedia Subsystem, IMS). La característica esencial del *middleware*, implementado en un Entorno de Ejecución de Lógica de Servicio (Service Logic Execution Environment, SLEE), consiste en que IMS utiliza a SOA para integrar sus propios elementos software con componentes externos y de esta manera, se logra

la combinación de las facilidades de la Web y de IMS para exponer un conjunto de servicios enriquecidos para ambos mundos.

Palabras clave: IT, IMS, SOA, middleware, SLEE.

ABSTRACT

The current trend in telecommunications is oriented to a search of a convergence of fixed and mobile networks, and therefore networks designed would be more complex. Also there are new challenges in the service networking and integration field across multiple networks, technologies and business areas, which involve there is a need to interoperate the Information Technologies (IT) and Telecommunication services. Therefore to contribute by solving these challenges but given the current lack of a fully adequate converged telecommunications environment –both traditional and new services–, this paper presents a reference architecture to enable and deliver quickly new converged services to the IT and telecommunications worlds, through SOA-IMS-based services interaction. Implemented in a SLEE, the middleware essential feature is that IMS uses SOA to integrate its own software components with external components, and thus it is achieved the IMS and Web services facilities combination to present a rich set of both services.

Keywords: IT, IMS, SOA, middleware, SLEE.

INTRODUCCIÓN

Cuando los servicios de telecomunicaciones son dependientes de las tecnologías de la red subyacente, limitan su evolución, respecto de la exigencia que introduce la velocidad de cambio sin precedentes, de las tecnologías utilizadas para el aprovisionamiento de servicios en la Internet. En este sentido, se requiere de redes de telecomunicaciones con arquitecturas de prestación de servicios que permitan la habilitación y entrega de nuevos servicios, a un ritmo similar al impuesto por los drásticos cambios de las Tecnologías de la Información y las Comunicaciones (TIC). Adicionalmente, se requiere que los nuevos servicios sean soportados por cualquier red (existente o futura) [1].

Al comparar los Servicios de Telecomunicaciones con los basados en Internet, se observa que para los primeros aún es desafiante el ciclo de desarrollo, en términos de tiempo y costo, debido principalmente a la manipulación directa de protocolos muy especializados. En el caso de los segundos, este problema se ha solventado con el uso de *Middleware* típicamente representado por librerías en un lenguaje de programación determinado. Por lo tanto, para reducir el tiempo de salida al mercado de los Servicios de Telecomunicaciones y adicionarles capacidades propias de la red

de redes, la convergencia de estos dos mundos ha adquirido en los últimos años, un interés considerable en el entorno de las Telecomunicaciones [2].

En este nuevo contexto tecnológico, donde es preponderante la creación rápida de servicios, es más eficiente la estimulación del desarrollo de servicios de próxima generación con y por parte de expertos en IT, ya que generalmente en ese ambiente las aplicaciones se desarrollan a alto nivel, utilizando *middleware*, y no a bajo nivel, utilizando directamente protocolos de comunicación. De esta forma, los sistemas heredados e IMS podrán seguir proporcionando las capacidades de la red subyacente, y el desarrollador no necesitará tener conocimiento profundo sobre protocolos específicos de capas inferiores, lo cual finalmente le permite concentrarse en el acceso simple y de alto nivel a una capacidad de telecomunicación, utilizando técnicas comunes de programación, típicamente basadas en Servicios Web (Web Service, WS).

A partir de la necesidad de proporcionarles a los proveedores de servicios, una facilidad software que combine el mundo IMS, con el Protocolo de Inicio de Sesión (Session Initiation Protocol, SIP), de las telecomunicaciones y el mundo SOA de las IT, para el desarrollo, despliegue y la prestación rápida y eficiente de servicios convergentes SOA/IMS, tanto para redes existentes como futuras, en este artículo se propone como solución un *Middleware* denominado MIDDIS, que reduce la complejidad de las telecomunicaciones para el mundo de las IT, puesto que el desarrollador no requiere entender la tecnología de la red subyacente con la cual se construyó cada servicio y en consecuencia, el *Middleware* permite una ejecución rápida de los procesos asociados al desarrollo de cualquier tipo de servicio en el nuevo entorno de convergencia.

En la sección 1, se presentan los antecedentes en la convergencia de servicios IMS y Web, se ilustran las principales alternativas que existen para el manejo de sesiones en los Servicios Web, y se describen brevemente, las tecnologías más representativas en el área de los *middleware* java para telecomunicaciones. La sección 2 describe la arquitectura de referencia del *middleware* para la interacción de servicios basados en SOA e IMS, denominado MIDDIS, y sus subsistemas. En la sección 3, se presenta un ejemplo de uso de MIDDIS y su implementación de referencia. Finalmente, la sección 4 corresponde a las conclusiones.

1. ESTADO ACTUAL DEL CONOCIMIENTO

A continuación, se definen brevemente los conceptos de IMS y SIP, y se describen algunas de las alternativas desarrolladas para dar solución a la necesidad de la convergencia entre los Servicios Web y los Servicios de Telecomunicaciones basados en IMS. Finalmente, se presentan los conceptos de SIP Servlets y de los SLEE, por ser de gran importancia para el pleno entendimiento de MIDDIS.

1.1. IMS y SIP

La evolución del Sistema de Telecomunicaciones Móviles Universales (Universal Mobile Telecommunications System, UMTS), incorporó al mundo de las telecomunicaciones un nuevo subsistema para el desarrollo de servicios multimedia y usuario a usuario. Esta nueva arquitectura del 3GPP se denomina IMS y se basa en SIP para la señalización. El objetivo de IMS es permitir la creación de nuevas aplicaciones multimedia capaces de suministrar al usuario final, una experiencia de comunicaciones integrada, independiente del tipo de aplicación, método de acceso a la red e infraestructura de la misma, de modo que los usuarios finales tengan acceso a cualquier tipo de servicios (tradicionales o nuevos), desde cualquier sitio, en cualquier momento, y sobre múltiples tecnologías de acceso [3].

SIP [4] es un protocolo de señalización del nivel de aplicación diseñado para el establecimiento, modificación, mantenimiento y terminación de sesiones interactivas sobre redes IP para diferentes tipos de aplicaciones [5]. En el 3GPP, se determinó la utilización de SIP como protocolo para conectar la gran mayoría de los nodos IMS entre sí, y con el resto de los elementos que componen una red de próxima generación. IMS utiliza también a SIP para establecer sesiones de multimedia y voz en Internet. El perfil de SIP creado para IMS (SIP-3GPP), es uno de los más importantes en el ámbito de las telecomunicaciones, pues no sólo involucra a las redes móviles sino también a toda la industria de las telecomunicaciones y según los expertos, actualmente es el más apropiado para las NGN [6].

1.2. CONVERGENCIA SERVICIOS WEB/IMS

WSIP: Web Service SIP Endpoint for Converged Multimedia/Multimodal Communication over IP [7]. Presenta un enfoque de doble pila, para la convergencia de servicios de comunicaciones sobre IP. Cada nodo WSIP es tanto un nodo SIP como un nodo del Protocolo Simple de Acceso a Objetos (Simple Object Access Protocol, SOAP), que proporciona un entorno nativo y genérico para la integración de servicios. No obstante, el enfoque de doble pila de WSIP puede no ser posible en las plataformas de WS o ni siquiera deseable, debido a que al ser SIP parte de su pila doble de señalización, un nodo WSIP precisa implementar y tratar con muchos de los asuntos que se relacionan con este protocolo, como por ejemplo la gestión de las sesiones, que a su vez, implica que el desarrollador debe tener el conocimiento y la experiencia necesaria para el desarrollo de servicios basados en la señalización SIP. En MIDDIS, estos inconvenientes quedan subsanados con el mecanismo de integración de servicios basado en SIP y WS, ya que éste se encarga de procesar simultáneamente tanto los protocolos como las extensiones de SIP y los WS, respetando de esta forma la independencia de los nodos.

Kogrimo [8], [9], describe una solución para fusionar las aplicaciones basadas en SIP y SOAP que permite llevar a la red móvil, los servicios de valor añadido basados en los WS. La solución de Akogrimo está orientada a las infraestructuras de Servicios basadas en Grid. MIDDIS utilizó los conceptos definidos en Akogrimo para integrar a SIP y SOAP; sin embargo, los conceptos de la comunidad Grid no se utilizaron.

WIMS 2.0 [10], esta iniciativa de Telefónica, hace especial énfasis en la exposición de capacidades de Telecomunicación al mundo Web 2.0 a través de la API REST y un Servidor de Aplicaciones IMS (ASIMS), que implementa la capa de adaptación entre ambos mundos. MIDDIS por su parte, adapta las funcionalidades SIP-IMS al dominio SOA-WS, utilizando como elemento integrador un SLEE.

SIP Based Real-Time Web Services Communication Model [11]. Propone un Modelo de Comunicaciones para Servicios Web en Tiempo Real (Real-Time Web Services Communication Model, RT-WSCM), basado en SIP, diseñado con máquinas de estados, mejoradas para el manejo de las llamadas en el RT-WSCM. En MIDDIS, se aplica el concepto de *middleware*, utilizando motores asíncronos, basados en eventos, como los SLEE.

1.3. MANEJO DE SESIONES EN LOS SERVICIOS WEB

A continuación, se presentan las alternativas para el manejo de sesiones de los WS. Estas son importantes para MIDDIS porque evitan sobrecargar la gestión de sesiones a través de SIP.

WS-*. La Organización para el Avance de Estándares de Información Estructurada (Organization for the Advancement of Structured Information Standards, OASIS), desarrolló diferentes estándares para la gestión de sesiones en los WS: WS-Coordination [12], WS-AtomicTransaction [13], WS-BusinessActivity [14] y WS-Context [15, 16]. WS-Coordination: es el estándar base, dirigido hacia la coordinación de metadatos. En lugar de especificar cómo se puede coordinar los WS, proporciona un marco para la transferencia de información acerca de la coordinación. Puede ser utilizado en SOA sencillas. WS-Context: es necesario en SOA más complejas, así como en las Arquitecturas de Componentes de Servicios (Service Component Architecture, SCA), y en WS que utilizan REST. MIDDIS implementa el concepto de sesión en los WS a través de la especificación WS-Context.

Gestión de la Sesión y de las Transacciones en los Servicios Web, utilizando SIP. Dong et al [17], desarrolla un nuevo mecanismo para dar soporte a la Gestión de la Sesión en los WS utilizando el protocolo SIP, con lo cual se propone una solución simple para la gestión de transacciones para WS. MIDDIS no realiza la gestión de sesiones a través de esta solución por ser propietaria.

1.4. MIDDLEWARE JAVA PARA TELECOMUNICACIONES

SIP Servlets. Definen un modelo basado en contenedores, que es una extensión del bien conocido modelo de Servlets del Protocolo de Transferencia de Hipertexto (Hypertext Transfer Protocol, HTTP). Fueron diseñados para simplificar el desarrollo de aplicaciones basadas en SIP y por lo tanto, para incrementar su adopción. Actualmente, los Servidores de Aplicaciones SIP dan también soporte a los Servlets HTTP y a los Servicios Web [18]. Sin embargo, no soportan una entidad funcional que por defecto registre las interfaces de un servicio IMS, de forma tal que otro servicio pueda utilizarlo en el proceso de creación de su lógica de negocio.

JSLEE. Es el estándar de Java para el SLEE, y surge de la tendencia actual, en el dominio de las telecomunicaciones, hacia las plataformas abiertas, estandarizadas y basadas en componentes. Está diseñado como un modelo de componentes, especializado en aplicaciones orientadas a eventos y de forma asíncrona, para entornos de baja latencia y alto rendimiento, es decir, las características básicas de los servicios de telecomunicaciones, en los cuales las transacciones deben ser livianas y rápidas de concluir [19] [20]. La especificación [21] incluye un modelo de componentes, para estructurar la lógica de las aplicaciones de comunicaciones, como una colección de componentes orientados a objetos y reutilizables, y para la composición de estos componentes en servicios de mayor nivel y más sofisticados [22]. Entre sus aportes se resaltan: reduce el tiempo de salida al mercado y el costo de desarrollo mediante la utilización de estándares, hace posible entornos de múltiples vendedores, proporciona un marco para la prestación de servicios portables, y hace una abstracción de la infraestructura subyacente con la utilización de Adaptadores de Recursos (Resource Adaptors, RA) [23]. A partir de las consideraciones anteriores, se seleccionó a JSLEE como servidor de aplicaciones para MIDDIS.

2. MIDDIS

2.1. ESCENARIO DE MOTIVACIÓN

Para el desarrollo de servicios en un entorno donde IMS y SOA se traten como dos mundos separados, es posible construir una plataforma de aplicaciones IMS que dé soporte sólo a peticiones SIP, que se enlacen a un portal de servicios, y luego utilizar herramientas SOA para desarrollar la notificación de los servicios, según los criterios de la Web 2.0 [24]. Esto permite la utilización de la admisión y el control de los recursos de IMS, y así se proveen funciones de gestión de entrada a usuarios y recursos, con el fin de prevenir la congestión y las interrupciones de los servicios. Sin embargo, la principal y mayor desventaja de este enfoque radica en que no se puede seguir controlando los servicios cuando salen del dominio SIP; en este caso, los operadores de telecomunicaciones se convierten en simples transportadores de bits, al no controlar la

señalización de los mismos, lo cual causa un impacto en extremo negativo a su modelo de negocio actual y futuro, ya que en ese contexto, no serían parte fundamental de la cadena de valor del nuevo mercado de servicios convergentes.

MIDDIS a diferencia del anterior enfoque, propone el desarrollo de servicios en un entorno donde IMS-SIP se pueden integrar con SOA-WS, con lo cual se busca que IMS llegue a representar realmente la estandarización de una SOA diseñada para aplicaciones de tiempo real, de gran escala, y segura [25]; propendiendo siempre por la participación de los operadores de telecomunicaciones en el control de la lógica de los servicios y por lo tanto, buscando mantener su relevancia en la cadena de valor de los servicios convergentes.

2.2. ARQUITECTURA DE REFERENCIA

Teniendo en cuenta los antecedentes en la convergencia IMS/SOA, se define a continuación la propuesta de integración de estas dos plataformas, a través de MIDDIS. Sus características básicas son:

- La provisión de una Pasarela SIP/SOAP que extiende las funcionalidades y capacidades en los procesos de señalización y control de servicios de ambos protocolos;
- La provisión de una Pasarela de Registro de Servicios IMS/WS por medio de la adaptación de los Registros de los servicios IMS a Registros únicos de Servicios Web;
- La provisión de una Pasarela de Medios IMS/WS que realiza la adaptación de los medios de Internet (tipos de medios Internet: RTSP, HTTP, etc.), a los de IMS (como el Protocolo de Transporte en Tiempo Real (Real-Time Transport Protocol, RTP), y el Protocolo de Control del Transporte en Tiempo Real (Real-Time Transport Control Protocol, RTCP), para el transporte del flujo IP multimedia en el plano del usuario).

A partir de las características anteriores, MIDDIS permite que clientes con capacidades IMS tengan acceso a los servicios prestados por los WS o viceversa; logrando así, la creación y el acceso a servicios IMS/SOA convergentes.

En la figura 1, se puede observar que el Subsistema de Interfaces de Recursos de Red (SSIRR), se comunica por una parte, con el Núcleo de IMS, lo cual hace a través de la señalización SIP, y, por la otra, se comunica con el cliente IMS; en este caso a través de SIP y RTP/RTCP. Así mismo, este subsistema provee al componente SOA/WS, que representa la arquitectura de un WS y está ubicado en el plano de Aplicaciones de la red IMS, las interfaces para el Lenguaje de Descripción de Servicios Web

(Web Services Description Language, WSDL), HTTP, SOAP y para la tecnología de Descripción, Descubrimiento e Integración Universal (Universal Description, Discovery and Integration, UDDI). Adicionalmente, el SSIRR se comunica con el Subsistema de Transmisión de Eventos (SSTE), cada vez que ocurre un evento proveniente de las interfaces de los recursos de la red subyacente, para que este lo envíe a los demás subsistemas.

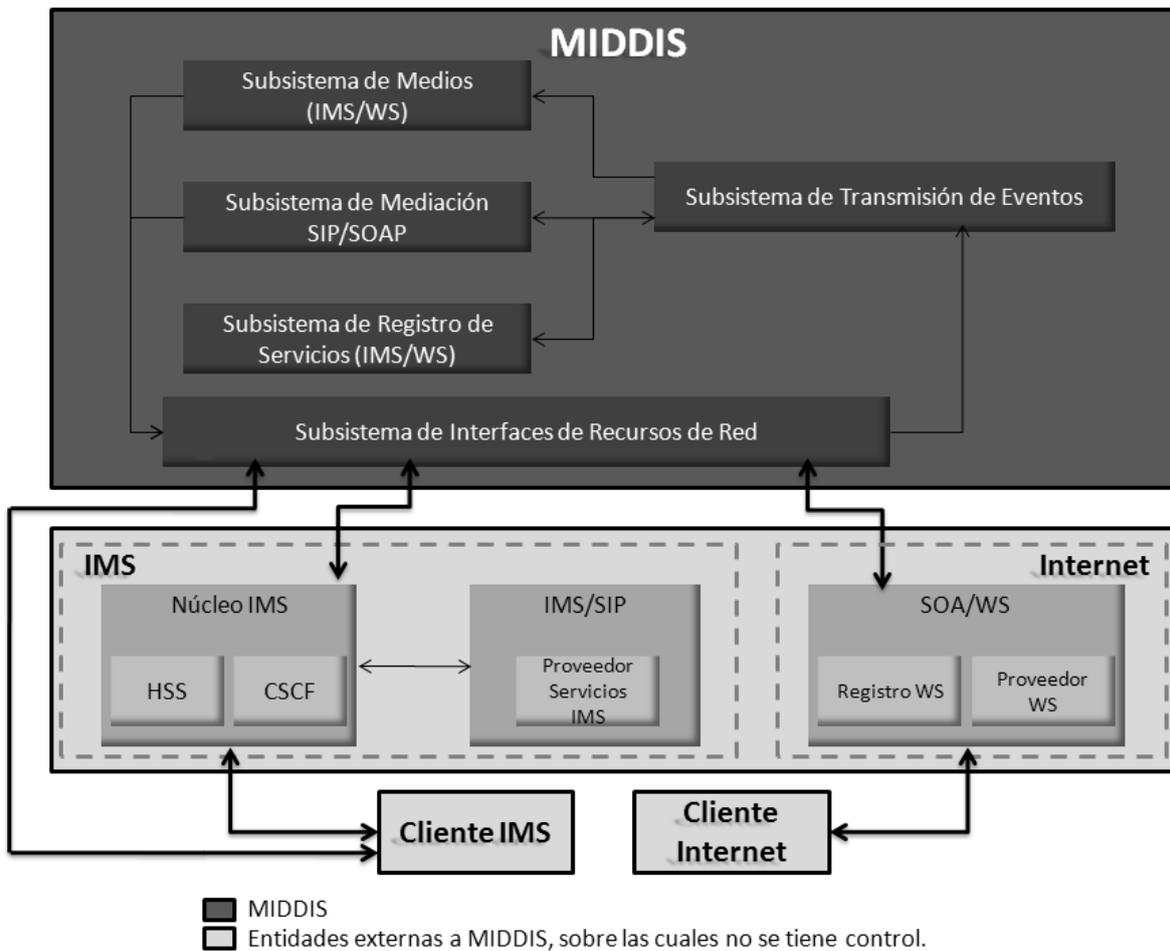


Figura 1. Arquitectura de Referencia

De manera inversa, el subsistema de Mediación SIP/SOAP (SSMIDD), y el Subsistema de Medios IMS/WS (SSMED), por una parte, envían al SSIRR los resultados de los procesos de mediación que deben ser transmitidos a las entidades externas a MIDDIS, y por la otra, generan peticiones intermedias que pueden ser enviadas en forma de eventos hacia los subsistemas restantes, utilizando el SSTE. A continuación, se describe brevemente cada subsistema de MIDDIS:

2.2.1. Subsistema de interfaces de recursos de red

Conecta el *Middleware* a la infraestructura de red subyacente, proporcionando las interfaces adecuadas para gestionar (suscribir y arbitrar), los eventos provenientes de esta última. La infraestructura de red subyacente corresponde específicamente a los elementos del núcleo de IMS y al cliente respectivo, por una parte, y al componente SOA/WS de provisión de WS, ubicado en la capa de aplicación de IMS, por la otra. Al recibir los eventos, este subsistema los adapta para convertir los protocolos y eventos específicos de la red en eventos genéricos en un lenguaje de programación específico, definiendo un significado equivalente. Además, gestiona los contextos de los flujos de eventos relacionados y envía los eventos adaptados hacia el SSTE. En sentido contrario, este subsistema recibe y maneja las solicitudes hechas por los subsistemas encargados de los procesos de Mediación (SSMIDD y SSMED), con el fin de generar respuestas hacia la red subyacente, para lo cual adapta los eventos genéricos del lenguaje de programación específico en eventos específicos del recurso de red al cual se dirigen los resultados de la mediación.

2.2.2. Subsistema de transmisión de eventos

Asegura y gestiona: i) El contexto en donde se realiza la transacción de cada evento y la correcta transmisión del mismo, direccionándolo al subsistema de Mediación interesado. ii) El contexto y la correcta transmisión de los eventos entre el SSMIDD y el subsistema de Registro de Servicios IMS/WS (SSREGS), que se requiere en los procesos intermedios de adaptación de la petición de registro de un servicio IMS en UDDI, antes de generar el resultado final. En este sentido, el SSTE es el corazón del direccionamiento de eventos del *Middleware*, y permite que los subsistemas de Mediación se comuniquen entre sí, y con la infraestructura de recursos subyacente.

2.2.3. subsistema de mediación sip/soap

Realiza la interoperación a nivel de señalización y control entre SIP y SOAP. Este subsistema juega el papel de *middleware* y controlador que ejecuta la adaptación entre el plano de señalización de IMS y los WS. Además, proporciona el control de los servicios y las funciones necesarias para la mediación en el registro de servicios convergentes IMS/WS.

Para lograr la gestión de la sesión de servicios basados en SIP/SOAP, y de acuerdo con el estado actual del conocimiento en MIDDIS, para el caso de SIP, se adicionan cabeceras, únicamente para la interoperación entre el Núcleo de IMS y el *Middleware*, con la información relacionada con los WS a los cuales se desea tener acceso. De

esta manera, se da el soporte a la negociación de sesiones SIP en donde ahora se intercambian nuevos tipos de medios y nuevos formatos de datos.

A pesar de que el protocolo SIP carece de mecanismos de control de servicios, puesto que en su lógica sólo comprende la gestión de sesiones, existen diversos mecanismos mediante los cuales los sistemas de comunicación multimedia basados en SIP, pueden intercambiar mensajes de control de servicios [26]. Por lo tanto, para generar los comandos de control de los servicios convergentes IMS/WS, en MIDDIS se optó por la extensión del uso de la petición MESSAGE, que inicialmente se definió para mensajería instantánea, utilizando el encabezado de tipo de contenido de la siguiente manera: Content-Type: application/soap+xml.

Por último, para proporcionar un paradigma de comunicaciones basado en WS con características típicas de *middleware*, y con capacidades para interactuar eficazmente con el protocolo de señalización SIP de IMS, MIDDIS extiende la pila básica de protocolos de los WS, acogiendo la especificación WS-Context [27], para adicionar la capacidad de manejo de sesiones en el dominio de los WS, y con ello mejorar la tarea del SSMIDD.

2.2.4. Subsistema de registro de servicios (IMS/WS)

Mediante este subsistema, se lleva a cabo la integración de los mecanismos de registro de servicios que existen tanto en WS/SOAP como en las redes IMS/SIP, y se llega a un solo mecanismo de almacenamiento de servicios, basado en el modelo de registro UDDI de los WS. Así, tanto proveedores como consumidores tienen acceso a los procesos de publicación y descubrimiento de servicios convergentes IMS/WS, de manera unificada.

2.2.5. subsistema de medios (IMS/WS)

Muchos de los formatos de datos y tipos de medios Web (como el Protocolo de Transmisión en Tiempo Real (Real-Time Streaming Protocol, RTSP), HTTP, incluyendo la descarga de archivos), no están soportados actualmente por las Pasarelas de Medios o por las Funciones de Recursos Multimedia de IMS. Además, los Agentes de Usuario SIP, tanto clientes como servidores, tampoco tienen la capacidad de tratar con muchos de ellos. En consecuencia, para asegurar la interacción entre el plano de medios de IMS/SIP y el de WS/SOAP, el principal propósito del SSMED es hacer posible que se recupere y reciba el contenido Web, proveniente de los Servidores de Aplicaciones, se adapte a los formatos y protocolos de medios de la red IMS, y se envíe hacia el Cliente IMS. El proceso inverso de adaptación no es crítico ya que el entorno Web es mucho más amplio en el soporte a formatos de datos y protocolos.

3. CASO DE ESTUDIO

3.1. REGISTRO DE UN SERVICIO WEB EN IMS

En la figura 2, se presenta el diagrama de secuencia que muestra el flujo de comunicación que se lleva a cabo para el Registro de un WS en IMS por medio de MIDDIS. Así, el WS estará disponible para el dominio del operador de telecomunicaciones.

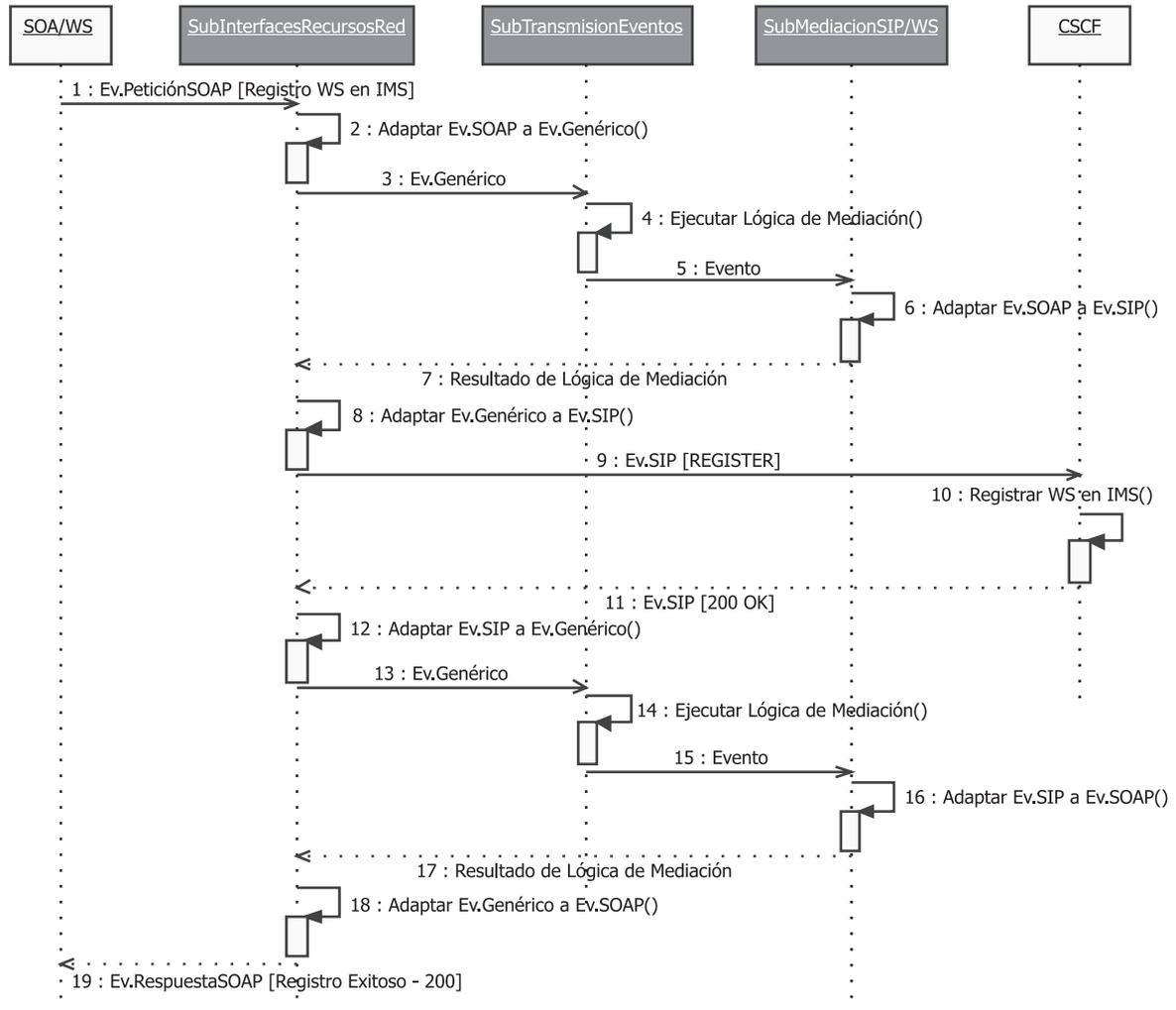


Figura 2. Registro de un Servicio Web en IMS

Descripción:

- El componente SOA/WS envía al SSIRR un Evento de Petición SOAP con la información necesaria para el registro del WS en IMS (URI de la WSDL del WS),

que realiza la adaptación del evento SOAP a un evento genérico java, y lo envía al SSTE (pasos 1-3).

- El SSTE determina cuál subsistema está interesado en el tipo de evento recibido. En este caso, al tratarse de un evento de Petición del tipo SOAP, lo dirige al SSMIDD que lo recibe y hace la adaptación del evento de tipo SOAP a un evento de tipo SIP que permita el registro del WS en IMS. Finalmente, el resultado de la mediación es enviado al SSIRR (pasos 4-7).
- El SSIRR recibe el resultado del proceso de mediación representado por un evento genérico en java, lo adapta a un evento específico de Petición SIP del tipo REGISTER, y lo entrega al componente de Funciones de Control de Sesión de la Llamada (Call Session Control Functions, CSCF), ubicado en el plano de control de la sesión de la red IMS. El CSCF ejecuta el proceso de registro del WS en IMS para que su uso esté disponible en los procesos de creación de los servicios convergentes IMS/SOA, y envía al SSIRR un Evento de Respuesta SIP del tipo 200 OK. El SSIRR realiza la adaptación del evento SIP a un evento genérico en java, y finalmente lo envía al SSTE (pasos 8-13).
- El SSTE redirige el evento al SSMIDD, que realiza la adaptación del evento de tipo SIP a un evento de tipo SOAP que permita confirmar el registro exitoso de un WS en IMS. El resultado de la mediación es enviado al SSIRR que lo adapta a un evento específico de Respuesta SOAP, y finalmente lo entrega al componente SOA/WS (pasos 14-19).

3.2. INVOCACIÓN DE UN SERVICIO WEB DESDE IMS

Para ilustrar el funcionamiento de la arquitectura de referencia de MIDDIS, se realizó la implementación de la invocación de un sitio Web de comercio electrónico, de venta de zapatos, desde un servicio IMS. El WS creado, Wsfootwearshop, permite a los usuarios la consulta, selección de artículos, y posterior compra de los mismos.

El modelo de implementación general incluye: el Agente de Usuario (User Agent, UA), IMS, que representa al usuario final del servicio; el CSCF, elemento encargado de la invocación del Wsfootwearshop; a MIDDIS que proporciona la mediación entre las interacciones con las capacidades del terminal IMS-SIP (UA IMS), y con el WS; y el componente SOA/WS que provee el Wsfootwearshop.

En el diagrama de secuencia que se muestra en la figura 3, se describe el proceso de envío de una orden de compra desde un UA IMS hacia el Wsfootwearshop, a través de MIDDIS. Una vez registrados tanto el terminal del UA IMS como el WSfootwearshop en la red IMS e iniciada la sesión entre los mismos, el servicio convergente IMS/WS provee al UA IMS, el acceso al WS por medio de la invocación de las operaciones de consulta, selección de artículos, y procesamiento de orden de compra, enviadas como eventos SIP a MIDDIS, a través del CSCF.

Para el caso de invocación de la funcionalidad de procesamiento de una orden de compra, el servicio convergente IMS/WS solicita al CSCF el envío de la misma a MIDDIS, la cual está contenida en el cuerpo de un mensaje SIP MESSAGE (con la URI de la WSDL del WS, el nombre de la operación por invocar, y el parámetro que contiene la orden de compra). MIDDIS procesa este mensaje de tipo SIP, y obtiene del cuerpo los datos de invocación al WS, y los adapta a un mensaje de tipo SOAP, que es enviado al WSfootwearshop. Este último procesa la orden de compra, envía de regreso una respuesta SOAP con los datos del envío de los artículos, y MIDDIS se encarga de adaptarla a una respuesta de tipo SIP 200 OK, en cuyo cuerpo se incluye el contenido del mensaje SOAP que finalmente se entrega al terminal UA IMS.

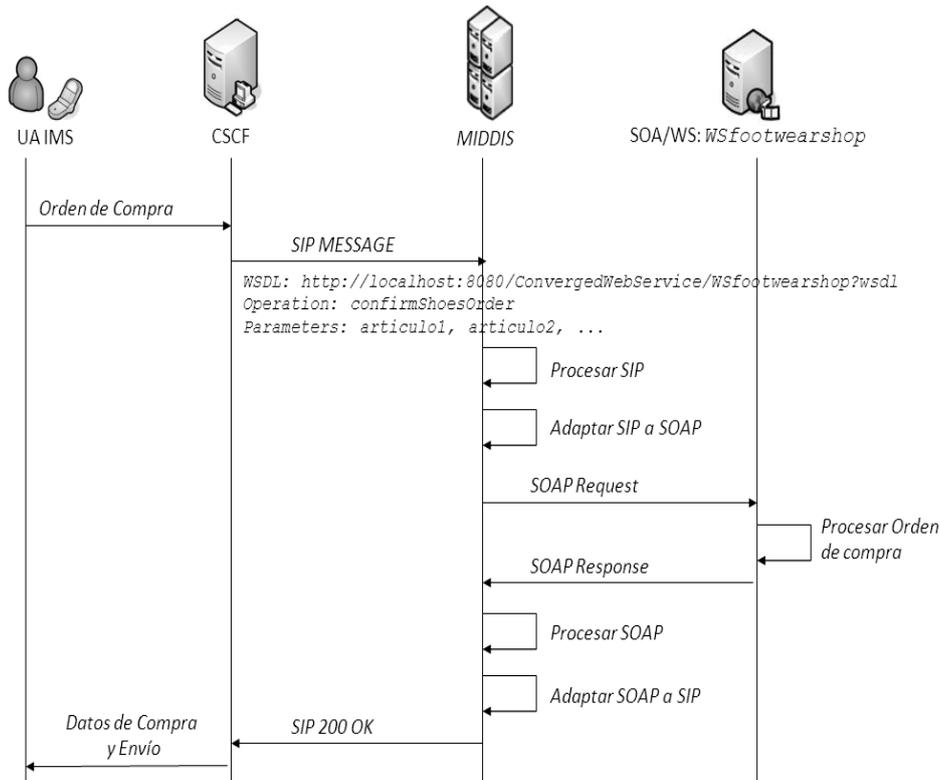


Figura 3. Invocación de un Servicio Web desde IMS

3.3. IMPLEMENTACIÓN DE REFERENCIA

Esta sección describe a través de los diagramas de paquetes y de despliegue, las tecnologías utilizadas para la implementación de MIDDIS.

3.3.1. Modelo de Implementación

En la tabla 1, se describen los componentes funcionales y en la figura 4, se presenta un esquema de los paquetes con los cuales se implementó MIDDIS.

Tabla 1. Descripción de los Componentes Funcionales del Middleware

COMPONENTE FUNCIONAL	DESCRIPCIÓN
Servidor JAIN SLEE	Contiene los módulos de adaptación a recursos externos y los bloques de lógica de servicio de MIDDIS.
IMS	Arquitectura de servicios IMS que comprende las funciones lógicas en cargadas del Acceso y Transporte, el Control de la sesión (núcleo IMS), y de las Aplicaciones (IMS/SIP, SOA/WS).

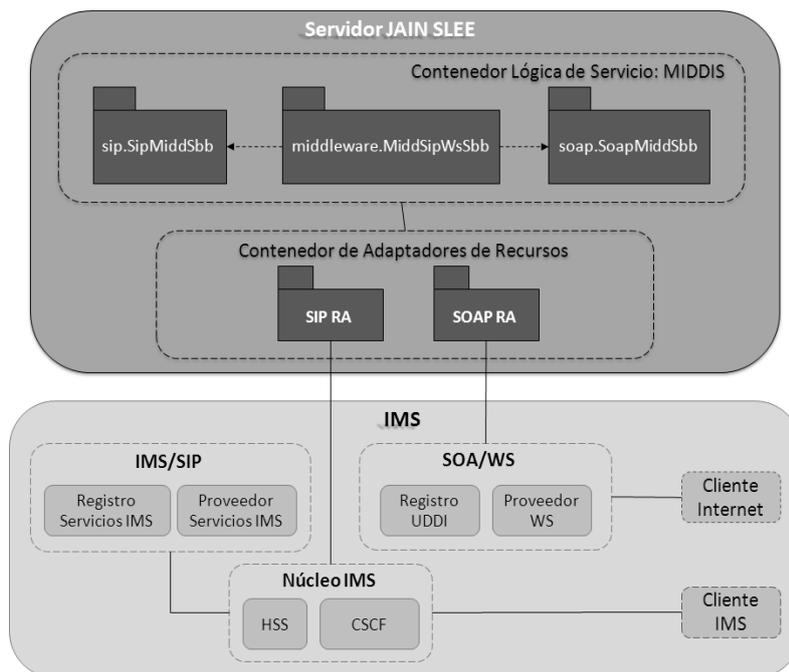


Figura 4. Paquetes de Diseño de MIDDIS

Entre las plataformas disponibles para el desarrollo de servicios JSLEE, se eligió a Rhino de OpenCloud [28], porque, entre otras cosas, ofrece el mayor portal con información referente a esta tecnología, es considerada la mejor plataforma de JSLEE [29], y en el momento de implementar a MIDDIS, era la única plataforma que cumplía completamente con la versión 1.1 de la especificación de JSLEE (1.1) [21], que permite obtener una mayor transparencia de la red subyacente a través de los RA.

La implementación de los SSIRR y SSTE de MIDDIS se realizó, a través de dos elementos constitutivos de la arquitectura de JSLEE: el plano de RA y del Enrutador de Eventos (que hace parte del marco de trabajo de JSLEE), respectivamente, debido a la correspondencia en características y funcionalidades entre dichos elementos.

Para la creación de la lógica de servicio de MIDDIS, conformada por los subsistemas restantes: SSMIDD, SSMED y SSREGS, se utilizó la FSM Tool, versión 0.8.5 [30], de OpenCloud, que es una herramienta liviana, orientada a simplificar la creación de servicios para Rhino SLEE. Desde la perspectiva del desarrollador, la herramienta se asegura de que los artefactos de la especificación y la implementación permanezcan completamente precisos y sincronizados durante todo el desarrollo y el mantenimiento del ciclo de vida de cada componente. Además, éste no tiene que codificar a mano y además mantener una jerarquía compleja de las Máquinas de Estados Finitos (Finite State Machine, FSM), que conforman el sistema.

Las FSM de los servicios que proporciona MIDDIS se definieron formalmente en un Lenguaje de Dominio Específico (Domain-Specific Language, DSL) basado en texto. Los principales pasos que se siguieron para el desarrollo de cada uno de los servicios de MIDDIS fueron:

1. Escribir la especificación de cada FSM, de manera que describiera completamente el comportamiento de los protocolos de comunicaciones SIP y SOAP. El modelado FSM de un protocolo de comunicaciones puede ser descrito como una tupla $(\Sigma, \Gamma, S, s_0, \delta, \omega)$, donde Σ es el alfabeto de entrada, Γ es el alfabeto de salida, S es un conjunto no vacío y finito de estados, s_0 es el estado inicial (pertenece a S), δ es la función de la transición $\delta: S \times \Sigma \rightarrow S$, es la función de salida $\omega: S \times \Sigma \rightarrow \Gamma$ [31].
2. Especificación de una FSM en un archivo de texto en el cual se definen los estados, las acciones, las transiciones, las entradas y las salidas. Los estados definidos para la FSM de cada protocolo de comunicaciones son:
SIP: INITIAL, REGISTERED, PROCEEDING, CONNECTED, TERMINATED, ERROR.
SOAP: INITIAL, PROCESSING, CONVERGED, FINAL, ERROR.
3. La FSM Tool utiliza la especificación FSM para generar una clase java del Bloque de Construcción de Servicios (Service Building Block, SBB), de la FSM.
4. Luego se extendió directamente el modelo de comportamiento especificado, es decir, la clase java del SBB de la FSM generada en el paso 2, adicionando implementaciones de las acciones definidas en la especificación de la FSM.
5. Se compiló, empaquetó y desplegó a MIDDIS, utilizando los métodos estándar para el despliegue completo de los SBB. El ejecutable del software creado consiste en los archivos fuente, las librerías, la unidad desplegable, y el descriptor del despliegue del servicio.

MIDDIS se dividió en tres paquetes que contienen los siguientes SBB:

- *sip.SipMiddSbb*: es un UA SIP que maneja toda la lógica de la señalización SIP, y representa al componente SOA/WS, específicamente al WSfootwearshop, dentro de la red IMS. Por una parte, recibe del SIP RA eventos SIP provenientes del

núcleo IMS, y por la otra recibe órdenes de generación de eventos SIP desde el MiddSipWsSbb, para ser enviados, a través del SIP RA, al núcleo IMS.

- *soap.SoopMiddSbb*: maneja toda la lógica de la señalización SOAP. Por una parte, recibe eventos del SOAP RA, provenientes del WSfootwearshop. Por la otra, recibe órdenes de generación de eventos SOAP desde el MiddSipWsSbb, para ser enviados al WSfootwearshop, a través del SOAP RA.
- *middleware.MiddSipWsSbb*: es el *Middleware* para la Interacción de Servicios basados en IMS (SIP) y SOA (SOAP); por lo tanto, implementa las funcionalidades de los subsistemas: SSMIDD, SSREGS, y SSMED. Este SBB adapta todas las peticiones y respuestas de un protocolo de comunicación (por ejemplo: SIP), al otro protocolo de comunicación (por ejemplo: SOAP). Los eventos enviados y recibidos por este SBB, pueden provenir tanto del SipMiddSbb como del SoapMiddSbb.

3.3.2. Modelo de Despliegue

La figura 5 ilustra la estructura de los nodos locales y remotos que constituyen MIDDIS.

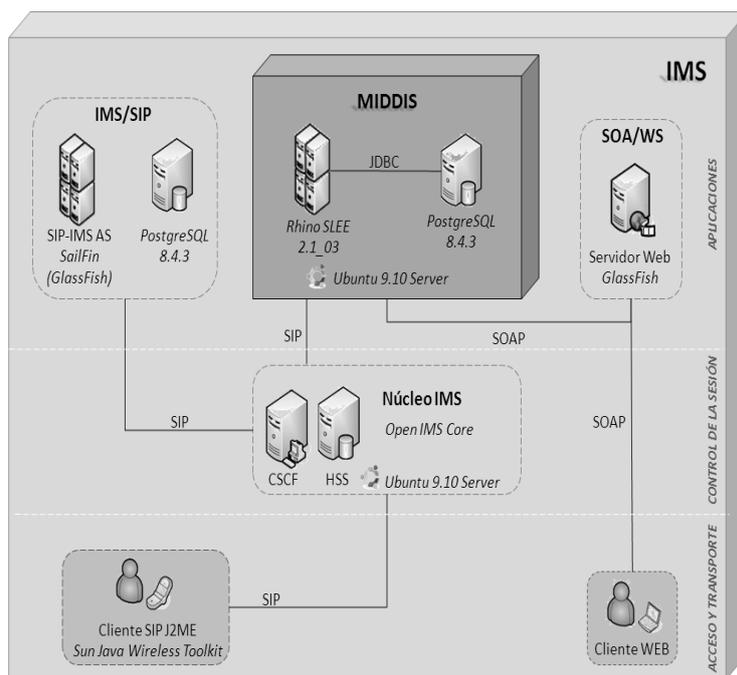


Figura 5. MIDDIS, diagrama de despliegue

- *MIDDIS*: se implantó sobre Rhino SLEE versión 2.1_03 de OpenCloud (con licencia académica), con las versiones 2.2_06 del SIP RA, y 2.1 del SOAP RA. La instalación se realizó en el OS Ubuntu 9.10 Server, y la Máquina Virtual Java

(JVM, Java Virtual Machine), versión 1.6.0_20. El motor de base de datos instalado para este ambiente de ejecución fue PostgreSQL 8.4.3, al cual se accedió a través de JDBC.

- *Núcleo IMS*: nodo que simula la red IMS. La Comunidad del Instituto Tecnológico Fraunhofer de Berlin para la investigación y desarrollo de sistemas de comunicaciones móviles, de código abierto, en redes fijas e inalámbricas, FOKUS (Fraunhofer institute Für Offene Kommunikations Systeme), implementó el Open IMS Core, que a su vez está inscrito dentro del proyecto “Open IMS Playground @ FOKUS”, un marco de desarrollo de aplicaciones para la tecnología IMS, dentro del cual se puede encontrar diferentes tecnologías de acceso, además de la implementación de referencia de todos los componentes del núcleo de IMS y las herramientas para su gestión. Por lo tanto, el Open IMS Core instalado proporciona la CSCF, los elementos centrales para el enrutamiento de la señalización IMS, y el Servidor Local del Suscriptor (Home Subscriber Server, HSS), para gestionar los perfiles de usuario y las reglas de enrutamiento asociadas [32].
- *IMS/SIP*: nodo que provee las aplicaciones SIP-IMS. Se implantó sobre SailFin (versión 2.0), que es un contenedor de SIP Servlets que corre sobre el servidor de aplicaciones Glassfish (versión 2.1), de Sun.
- *SOA/WS*: nodo que provee los Servicios Web. Se implantó sobre Glassfish versión 2.1.

4. CONCLUSIONES

MIDDIS maneja la convergencia de servicios de interacción bajo un enfoque arquitectónico que incluye aspectos tanto de SOA como de IMS. En este sentido, se basa en la naturaleza versátil de SIP y de SOAP, los extiende y complementa para integrar a IMS con SOA, en su capa de aplicaciones, y con ello proporciona una gran variedad de oportunidades en la creación de servicios convergentes. De esta manera, por una parte el entorno resultante ahora se centra en el usuario, y no en los servicios, y por la otra, se proporciona un medio adecuado para la prestación de los servicios necesarios para justificar las inversiones en IMS.

En MIDDIS, se considera la utilización de la interacción SIP/SOAP para transmitir tanto los mensajes de señalización como los mensajes de control, y no solamente la utilización de SIP para la ejecución de estas dos tareas. En otras palabras, el manejo de sesiones y el control de servicios por medio de SIP en IMS, se complementa a través de MIDDIS, con el manejo de sesiones por medio de WS-Context, y el control de servicios en los WS, con los mensajes SOAP. Por lo tanto, MIDDIS proporciona el manejo de sesiones y el control de servicios para ambos entornos, lo cual implica a su vez, mayores posibilidades a la hora de realizar estos dos procesos en el ámbito de los servicios convergentes IMS/SOA. De esta manera, se facilita en gran medida la

ejecución de estas tareas y se aporta, consecuentemente, mayores beneficios tanto para operadores de red como para proveedores de servicios, ya que los primeros no pierden importancia dentro de la cadena de valor, y los segundos pueden acceder con más facilidad a las capacidades de ambos entornos.

MIDDIS constituye una arquitectura de referencia, basada en tecnologías abiertas y estandarizadas para la mediación en la interacción de servicios basados en IMS y SOA, y por lo tanto, para la creación rápida de servicios convergentes IMS/SOA. Proporciona los servicios de adaptación SIP/SOAP, basados en JSLEE, para los procesos de: Registro de un WS en IMS; inicio, mantenimiento y finalización de sesión entre un servicio IMS y uno Web; y acceso a las funcionalidades de un WS desde un servicio IMS.

Los WS no se identifican, descubren y localizan por medio de direcciones SIP; sin embargo, la interacción entre servicios basados en IMS y SOA se puede realizar mediante la iniciación de sesiones SIP, a través de MIDDIS.

Se define y ejecuta el comportamiento de los protocolos de comunicaciones SIP y SOAP a través de Máquinas Virtuales de Estados Finitos. La importancia del uso de las FSM es que éstas se pueden extender a través de FSM más complejas, manteniendo la independencia entre el modelo y el código de implementación.

REFERENCIAS BIBLIOGRÁFICAS

- [1] AePONA (2005). VAS Implementation in the IMS. Consultada en octubre del 2010. En: <http://www.azouk.com/216436/Aepona-IMS-White-paper/>
- [2] Tarkoma, S., Rovira, J., Postmann, E., Rajasekaran, H., Kovacs, E. (2008). Creating Converged Services for IMS Using the SPICE Service Platform. Consultada en marzo de 2008. En: <https://www.icin.biz/files/programmes/Session7B-3.pdf>
- [3] Expocomm Argentina. (2005). Seminario de Tecnología y Mercado. Consultada en abril del 2005. En: http://www.ejkreed.com/extras/img/web_727_brochurestm.pdf
- [4] Rosenberg, J., y otros. (2002). SIP: Session Initiation Protocol. Consultada en abril de 2005. En: <http://www.rfc-editor.org/rfc/rfc3261.txt>
- [5] Telefónica I+D. (2005). Las Telecomunicaciones y la Movilidad en la Sociedad de la Información. Consultada en abril del 2006. En: http://sociedadinformacion.fundacion.telefonica.com/docs/repositorio//es_ES//TelefonicaySI/Publicaciones/telecoymovilidad.pdf

- [6] RADVISION. (2006). IMS SIP and Signaling, The RADVISION Perspective. Consultada en agosto del 2006. En: <http://www.radvision.com/NR/rdonlyres/FC60D840-1FE5-4F82-A6A2-088D2D4AADCB/0/IMSSIPWhitePaper.pdf>
- [7] Liu, F., Chou, W., Li, L., Li, J. (2004). WSIP – Web Service SIP Endpoint for Converged Multimedia/Multimodal Communication over IP. Consultada el 29 de agosto del 2008. En: <http://dpmn.postech.ac.kr/mcs/papers/ALR-2004-004-paper.pdf>
- [8] Jähnert, J., Cuevas, A., Moreno, J. I., Villagrà, V. A., Wesner, S., Olmedo, V., Einsiedler, H. The “Akogrimo” way towards an extended IMS architecture. Consultada en enero de 2009. En: <http://www.icin.biz/files/programmes/Session2A-1.pdf>
- [9] Villagrà, V. A., Wesner, S. AKOGRIMO Mobile Grids: Mobile Dynamic Virtual Organizations. Consultada en mayo de 2009. En: http://www.coregrid.net/mambo/images/stories/Events/BIGG/session%205_villagra.pdf
- [10] Moro Fernandez, D., Lozano Llanos, D., y Galindo Sánchez, L. A. (2008). WIMS 2.0: la convergencia del mundo Telco con la web 2.0. Consultada en octubre del 2010. En: http://sociedadinformacion.fundacion.telefonica.com/DYC/SHI/seccion=1188&idioma=es_ES&id=2009100116310161&activo=4.do?elem=7518
- [11] Cheng, B., Guo, J., Meng, X., Chen, J. (2008). SIP Based Real-Time Web Services Communication Model. IEEE Computer Society, ISECS International Colloquium on Computing, Communication, Control, and Management, pp. 439-443. 2008.
- [12] WS-Coordination (2007). Web Services Coordination (WS-Coordination), Version 1.1. Consultada en mayo de 2007. En: <http://docs.oasis-open.org/ws-tx/wscoor/2006/06>
- [13] WS-AtomicTransaction (2007). Web Services Atomic Transaction (WS-AtomicTransaction), Version 1.1. Consultada en mayo de 2007. En: <http://docs.oasis-open.org/ws-tx/wsat/2006/06>
- [14] WS-BusinessActivity (2007). Web Services Business Activity (WS-BusinessActivity), Version 1.1. Consultada en mayo de 2007. En: <http://docs.oasis-open.org/ws-tx/wsba/2006/06>
- [15] WS-Context (2007). Web Services Context Specification (WS-Context) Version 1.0. Consultada en mayo de 2009. En: <http://docs.oasis-open.org/ws-caf/ws-context/v1.0/wsctx.html>

- [16] Dornan, A. (2007). Tech Road Map: Oasis Takes On Web Services Session Management. Consultada en mayo de 2009. En: <http://www.informationweek.com/news/software/soa/showArticle.jhtml?articleID=201804545>
- [17] Dong, W., Newmarch, J. Adding Session and Transaction Management to Web Services by using SIP. Consultada en marzo de 2008. En: <http://jan.newmarch.name/publications/sip-soap.pdf>
- [18] Kmatveev (2008). Java middleware for telecom: JSLEE vs. SIP Servlets. Consultada en enero de 2009. En: <http://technfun.wordpress.com/2008/03/03/java-middleware-for-telecom-jslee-vs-sip-servlets/>
- [19] Ivanov, I. (2006). Mobicents: JSLEE for the People, by the People. Consultada en enero de 2009. En: <http://today.java.net/pub/a/today/2006/03/09/mobicents-jslee.html>
- [20] Cruz, A. (2005). Una nueva convergencia: ¿Java en la red?. Consultada en enero de 2009. En: http://www.iworld.com.mx/iw_SpecialReport_read.asp?iwid=3827&back=2&HistoryParam=U
- [21] Sun Microsystems y OpenCloud (2008). JAIN SLEE (JSLEE) 1.1 Specification, Final Release. Consultada en mayo de 2009. En: <http://jcp.org/aboutJava/communityprocess/final/jsr240/index.html>
- [22] JAINSLEE.org, Universidad de Otago, OpenCloud, Harmonic. JAIN SLEE Fundamentals. Consultada en enero de 2009. En: <http://www.jainslee.org/slee/fundamentals.html>
- [23] Marezke, M. (2005). JAIN SLEE Technology Overview, Version 1.1. Consultada en enero de 2009. En: http://www.marezke.de/pub/lectures/JSLEE_Overview_2005/JSLEE_Overview_2005.pdf
- [24] Nolle, T. (2008). Building revenue-increasing telecom services for the future. Consultada en abril de 2009. En: http://searchtelecom.techtarget.com/tip/0,289483,sid103_gci1320561_mem1,00.html
- [25] McHugh, M. (2006). IMS & SOA — Driving the Future of Telecommunications. Consultada en agosto de 2008. En: <http://www.tmcnet.com/ims/1206/industry-perspective-ims-and-soa-driving-the-future.htm>
- [26] Almeida Cruz, Y. (2007). Plataforma para el establecimiento y desarrollo de conferencias multimedia. Consultada en mayo de 2009. En: <http://www.cujae.edu.cu/Eventos/CITTEL/Memorias/CITTEL2004/Trabajos/CIT007.pdf>

- [27] Newcomer, E. (2007). IONA. WS-Context moving thru OASIS standard process. Consultada en mayo de 2009. En: <http://blogs.iona.com/newcomer/archives/000442.html>
- [28] OpenCloud. (2010). Rhino 2.1-03 release. En: <https://developer.opencloud.com/devportal/display/OCDEV/Rhino+2.1>
- [29] Ruiz Rojas, J. (2009). Introducción a JAIN SLEE. Consultada en agosto de 2009. En: <http://www.slideshare.net/jruizro/developmentinjainsleemay2009?nocache=7891>
- [30] FSM Tool, OpenCloud. (2009). FSM Tool Developer's Guide. Consultada en mayo de 2009. En: <https://developer.opencloud.com/devportal/display/OCDEV/FSM+Tool>
- [31] Basicovic, L., Popovic, M., Velikic, I. (2010). Use of Finite State Machine Based Framework in Implementation of Communication Protocols – A Case Study. IEEE Computer Society, 2010 Sixth Advanced International Conference on Telecommunications, pp. 161-166. 2010.
- [32] Open IMS Core. En: <http://www.openimscore.org/>

