



Revista Ingenierías Universidad de
Medellín
ISSN: 1692-3324
revistaingenierias@udem.edu.co
Universidad de Medellín
Colombia

Caicedo, Julián Andrés; Corrales, Juan Carlos
Aproximación a un modelo de aprovisionamiento de servicios convergentes
Revista Ingenierías Universidad de Medellín, vol. 13, núm. 25, julio-diciembre, 2014, pp.
207-225
Universidad de Medellín
Medellín, Colombia

Disponible en: <http://www.redalyc.org/articulo.oa?id=75039185013>

- Cómo citar el artículo
- Número completo
- Más información del artículo
- Página de la revista en redalyc.org

Aproximación a un modelo de aprovisionamiento de servicios convergentes

*Julian Andrés Caicedo**
*Juan Carlos Corrales***

Recibido: 10/12/2013 • Aceptado: 27/06/2014

Resumen

El aprovisionamiento de servicios de telecomunicaciones ha cambiado notablemente en los últimos 30 años; se ha pasado de modelos estáticos, rígidos y controlados por un solo actor de negocio a modelos dinámicos, flexibles, con múltiples actores en la cadena de valor y orientados a los usuarios finales. Diferentes enfoques han planteado modelos de aprovisionamiento que cumplan requerimientos específicos a los operadores de telecomunicaciones. Sin embargo, aún no están claros los procesos principales que deben desarrollarse en cada fase del ciclo de vida del servicio para que permitan una integración efectiva entre diferentes modelos de aprovisionamiento. En este artículo se presenta una aproximación al modelo inicial de aprovisionamiento de servicios convergentes, el cual abstrae los principales procesos dentro de las fases de diseño, despliegue y operación del servicio. De igual manera, se presenta un piloto funcional que ejecuta de manera automática el proceso de despliegue inicial en un entorno convergente con base en los procesos definidos en esta fase. Finalmente, el mecanismo es sometido a una prueba de escalabilidad para evaluar su desempeño.

Palabras clave: servicios convergentes, JSlee, aprovisionamiento de servicios, mecanismo de despliegue automático.

* M.Sc(c) en Ingeniería Telemática de la Universidad del Cauca (2014). Miembro del Grupo de Ingeniería Telemática (GIT); Facultad de Ingeniería Electrónica y Telecomunicaciones, Departamento de Telemática, Grupo de Ingeniería Telemática, Universidad del Cauca, Popayán, Colombia. E-mail: jacacicedo@unicauca.edu.co

** Ph.D. Profesor Titular y Líder del Grupo de Ingeniería Telemática (GIT); Facultad de Ingeniería Electrónica y Telecomunicaciones, Departamento de Telemática, Grupo de Ingeniería Telemática, Universidad del Cauca, Popayán, Colombia. E-mail: jcorral@unicauca.edu.co

Towards a model of convergent service provisioning

Abstract

In the last thirty years telecom service provisioning has changed significantly. In the early years, there were static and rigid provisioning models and these were controlled by only one business actor. After that, a new flexible and dynamic provisioning models with a lot stakeholder in the value chain appeared in the telecom market. Approaches have proposed service provisioning models for telecom operator; however, it's not clear what process have to carry out in each services lifecycle phase in order to integrate different service provisioning models in an efficient way. In this paper, we present an approach to initial model of convergent services provisioning emphasizing in those generic processes inside of design, deployment and operation service. Moreover, we built an automatic mechanism for initial deployment in a convergent environment based on the process defined in that service phase. Finally, we validate that automatic mechanism with a scalability test.

Keywords: Convergent services, Jain Slee, Service provisioning, Automatic deployment mechanism

INTRODUCCIÓN

El aprovisionamiento de servicios de telecomunicaciones ha cambiado notablemente en los últimos 30 años; se ha pasado de modelos estáticos, rígidos y controlados por un solo actor de negocio, a modelos dinámicos, flexibles, con múltiples actores en la cadena de valor, y orientados a los usuarios finales. En un principio, se contaba con modelos tradicionales cerrados donde prevalecía el conocimiento técnico del mundo Telco [1]. Luego, con el objetivo de abrir la red, surge un modelo orientado a terceros, donde un proveedor de servicios de aplicación (ASP, Application Service Provider) externo puede hacer uso de las capacidades de red del operador de telecomunicaciones para prestar un servicio con nuevas y mejores funcionalidades [2]. Esta iniciativa (OSA/Parlay), luego convertida en estándar, se puede considerar como el punto de partida de un sistema de aprovisionamiento abierto de servicios de telecomunicaciones.

Posteriormente, con el surgimiento de nuevas organizaciones, estándares y tecnologías, se habilitó un escenario tecnológico para la creación de modelos de aprovisionamiento de servicios de telecomunicaciones orientados a usuarios finales, donde el usuario no es solo un consumidor del servicio, sino que es también, un creador de contenidos y aplicaciones [3].

De manera general, el aprovisionamiento de servicios de telecomunicaciones puede ser entendido como aquellas actividades, ejecutadas en secuencia correcta, que hacen disponibles los recursos necesarios para consumir un servicio [4]. Estas actividades se encuentran distribuidas principalmente sobre tres de las fases del ciclo de vida de un servicio: *diseño/desarrollo, despliegue y operación*.

Diferentes aproximaciones académicas y empresariales han enfocado sus estudios en cada una de las fases del ciclo de vida del servicio. Para la fase de composición, enfoques como [5-9] construyen herramientas o ambientes de creación de servicios; otros como [10-12] se enfocan en la técnicas de recuperación y composición dinámica, haciendo énfasis en alternativas semánticas. En la fase de despliegue, trabajos como [13-15] realizan un despliegue inicial del servicio, considerando en él solo las funciones o lógica principal, y dejando de lado actividades de configuración, activación e instalación, todas ellas necesarias para activar adecuadamente la fase de operación del servicio. Por último, la fase de operación tiene una relevancia empresarial alta, principalmente por ser esta la que prepara el servicio para su consumo final. Soluciones tecnológicas como la de IBM [16], Oracle [17] y Ericsson [18] proponen sistemas de operación para soportar correctamente el consumo de un servicio final. Con estos sistemas se pretende automatizar la operación y la administración de servicios de telecomunicaciones. En [19] se presenta un estudio comparativo entre los principales proveedores de estos sistemas.

No obstante, integrar dichas aproximaciones para cada una de las fases del ciclo de vida del servicio es complejo y tedioso, principalmente por el uso de tecnologías, en algunos casos, propietarias y con interfaces no estandarizadas. En consecuencia, los operadores de telecomunicaciones deben implementar módulos intermedios para lograr la interoperabilidad entre sistemas. Para enfrentar esta problemática, el TMF fórum en [20] ha establecido que cada una de las fases del ciclo de vida tenga interfaces estandarizadas con el fin de proveer la comunicación adecuada entre los diferentes sistemas de aprovisionamiento. En efecto, la heterogeneidad de estos sistemas se debe a la dinámica generada por los servicios avanzados de telecomunicaciones, donde la cadena de suministro está compuesta por varias actores de negocio, muchos de ellos, fuera del dominio interno del operador de telecomunicaciones (servicios web, proveedores de contenido, proveedores de aplicaciones, etc.) y con tecnologías de despliegue y ejecución diferentes.

Adicionalmente, aún no están claros los procesos que deben ejecutarse en cada fase del ciclo de vida del servicio. En ese sentido, se plantea en el presente artículo un modelo inicial para el proceso de aprovisionamiento de servicios convergentes, el cual abstrae las principales funciones dentro de las fases de diseño, despliegue y operación. Para ello, se parte de lo expuesto por Vanea Chiprianov [21] donde se exponen las etapas genéricas de las fases del ciclo de vida de un servicio de telecomunicaciones, y se complementa con los enfoques propuestos en [22-23]. El primero describe las actividades de la fase de despliegue para entornos virtuales; el segundo presenta un modelo de referencia en procesos de negocio para los proveedores de servicios de telecomunicaciones (eTOM, Enhanced Telecom Operations Map). De igual manera, se plantean algunas consideraciones de contexto, necesarias para el aprovisionamiento de servicios convergentes. Finalmente, se define un piloto para automatizar la fase de despliegue de servicios de telecomunicaciones en un entorno convergente con base en los procesos definidos para dicha fase, se evalúa el desempeño del piloto aplicando la prueba de escalabilidad, se presentan los resultados obtenidos y se exponen las conclusiones finales.

1. MATERIALES Y MÉTODOS

El modelo inicial para el aprovisionamiento de servicios convergentes definido en este artículo parte del concepto asociado a cada una de las fases del ciclo de vida de un servicio de telecomunicaciones y se complementa con aquellos procesos genéricos para cada una de las fases definidas [20-23]. Finalmente, se definen las consideraciones de contexto necesarias para un modelo de aprovisionamiento de servicios convergentes.

A. Modelo inicial de aprovisionamiento de servicios convergentes

El aprovisionamiento de servicios de telecomunicaciones emplea aquellas actividades, ejecutadas en secuencia correcta, que hacen disponibles los recursos necesarios para consumir un servicio [4]. En ese sentido, y teniendo en cuenta las fases del ciclo de vida de un servicio, definidas por el TeleManagement Forum [20], es posible inferir las fases que lo componen: *diseño-desarrollo*, *despliegue* y *operación*. Las dos primeras hacen parte del proceso de composición de servicios; la última, orientada al desarrollo de sistemas de operación y soporte (OSS, Operations support systems). La figura 1 ilustra las fases que componen el proceso de aprovisionamiento.

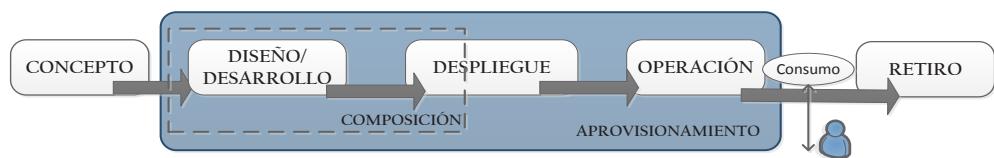


Figura 1. Fases Aprovisionamiento de Servicios de Telecomunicaciones

Fuente: Propia

- *Diseño/construcción*: fase que incluye aquellas actividades orientadas a la definición y desarrollo de las capacidades funcionales, no funcionales y de gestión del servicio.
- *Despliegue*: fase que abarca aquellas actividades orientadas en hacer que una instancia de servicio esté disponible para los usuarios o para ser combinada con otros servicios.
- *Operación*: fase que contempla aquellas actividades orientadas a soportar la ejecución del servicio y sus dependencias desde dominio del proveedor del servicio, del suscriptor del servicio, del proveedor de red y del usuario final.

Las anteriores fases son una abstracción del proceso de aprovisionamiento de un servicio de telecomunicaciones, independiente del dominio donde opere el servicio (web o Telco). Sin embargo, es necesario definir características de contexto para lograr un efectivo aprovisionamiento del servicio, específicamente para entornos convergentes. En consecuencia, el modelo propuesto en este trabajo unifica los procesos empleados para cada una de las fases y adiciona un componente de contexto, creando una estructura de referencia para operadores de telecomunicaciones o proveedores de aplicaciones que deseen implementar sistemas de aprovisionamiento de servicios convergentes. Es importante resaltar que la modularidad del modelo permite la construcción de mecanismos independientes para cada fase, facilitando el control y manipulación de los recursos.

Teniendo en cuenta [5], un servicio convergente es aquel que opera entre dos redes diferentes, particularmente, la red del operador y la web. En consecuencia, el aprovisionamiento de este tipo de servicios debe considerar aspectos de contexto de usuario, de dispositivo, de servicio y de seguridad y privacidad. A continuación se describe el modelo inicial para el aprovisionamiento de servicios convergentes; figura 2 ilustra el modelo.

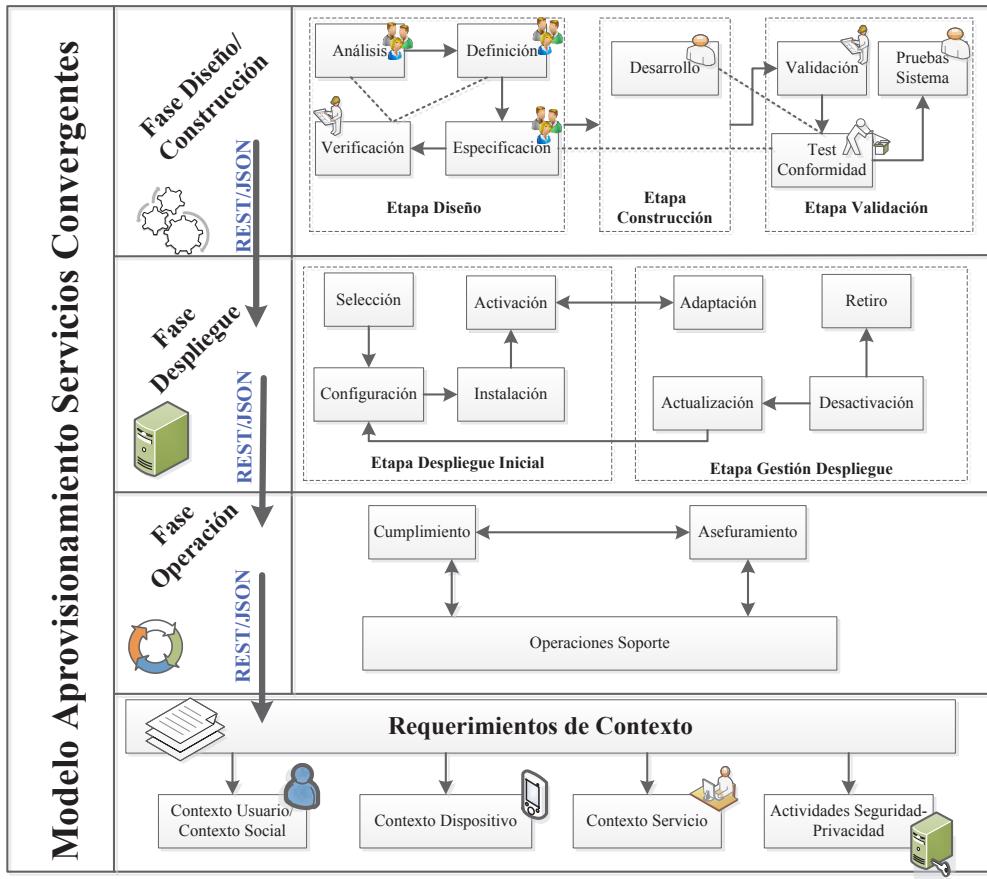


Figura 2. Modelo inicial de aprovisionamiento de servicios convergentes

Fuente: Propia

El modelo inicial de aprovisionamiento de servicios convergentes está dividido en tres capas principales: *fase de diseño*, *fase de despliegue* y *fase de operación*; todas ellas controlan el ciclo de vida del servicio, y están soportadas por la capa de *contexto*, la cual aborda los requerimientos convergentes en el aprovisionamiento del servicio. En consecuencia, este modelo describe aquellos procesos genéricos y necesarios para realizar el aprovisionamiento de servicios en un entorno convergente.

A.1 Fase diseño-construcción

Esta fase está compuesta de tres etapas. La primera incluye los procesos encargados de la definición del servicio; la segunda ejecuta los procesos de construcción funcional y no funcional del servicio; la tercera ejecuta los procesos encargados de la validación del servicio final con respecto a los requerimientos y políticas definidas.

A.1.1 Etapa de diseño

- *Análisis*: proceso encargado de describir de manera estructurada, ordenada y sistemática los requerimientos del servicio teniendo en cuenta las restricciones y políticas establecidas por los actores involucrados en el aprovisionamiento del mismo.
- *Definición*: proceso encargado de describir la meta de negocio del servicio para cada uno de los actores involucrados en el aprovisionamiento del servicio. De igual manera, se identifican las dependencias y relaciones involucradas en la cadena de aprovisionamiento como los elementos software o hardware, los recursos usados o los servicios empleados por el servicio definido.
- *Especificación*: proceso encargado de describir de manera formal e inequívoca el servicio. En este proceso se detalla la manera de implementar los módulos que lo componen y la interacción existente entre ellos.
- *Verificación*: proceso que analiza la correspondencia entre el proceso de especificación del servicio y los requerimientos iniciales (procesos de análisis y definición).

A.1.2 Etapa de construcción

- *Desarrollo*: proceso que comprende la definición e implementación de los módulos software y hardware necesarios para el funcionamiento del servicio especificado.

A.1.3 Etapa de validación

- *Validación*: proceso encargado de poner a prueba el servicio implementado con el fin de asegurarse de su correcto funcionamiento.
- *Test de conformidad*: proceso que valida la implementación del servicio con las reglas de arquitectura, restricciones y políticas definidas en el diseño. De esta manera, se verifica la correspondencia entre el resultado final del proceso de desarrollo y de los procesos de definición y de especificación.
- *Pruebas de sistema*: proceso encargado de ejecutar un test o banco de pruebas para los módulos software y hardware en un ambiente operacional.

A.2 Fase despliegue

Esta fase está compuesta de dos etapas. La primera incluye los procesos para soportar el despliegue inicial del servicio; la segunda ejecuta los procesos de gestión del servicio una vez esté activo.

A.2.1 Etapa despliegue inicial

- *Selección*: proceso encargado de seleccionar los objetos apropiados (software-hardware) para desplegar el servicio.
- *Activación*: proceso encargado de activar y hacer disponible el servicio para su ejecución.
- *Instalación*: proceso encargado de la adición de los componentes software al ambiente de despliegue.
- *Configuración*: proceso encargado de ajustar los componentes del servicio a las especificaciones del ambiente de despliegue.

A.2.2 Etapa de gestión de despliegue

- *Adaptación*: proceso encargado de adaptar los componentes software instalados y configurados previamente mientras el ambiente de despliegue está corriendo y recibiendo solicitudes.
- *Actualización*: proceso encargado de remplazar componentes software, previamente instalados, y reiniciar una configuración en ellos.
- *Desactivación*: proceso que hace uso de operaciones de mantenimiento para controlar aquellos módulos que no pueden operar en un servicio activo.
- *Retiro*: proceso que remueve los componentes software del servicio del ambiente de despliegue.

A.3 Fase de operación

Esta fase está compuesta por los procesos de *cumplimiento y aseguramiento*, (FA, Fulfillment, Assurance) y sus *operaciones de soporte*.

- *Cumplimiento*: proceso responsable de proveer, de manera oportuna y correcta, los productos solicitados por el cliente.
- *Aseguramiento*: proceso responsable por la ejecución de actividades proactivas y reactivas de mantenimiento para asegurar que el servicio provisto a los clientes

están continuamente disponibles y cumpliendo adecuadamente con los acuerdos de SLA y QoS.

- *Operaciones de soporte*: proceso responsable de la gestión, control y soporte administrativo para el correcto funcionamiento de FA, asegurando la efectividad en el cumplimiento de sus funciones.

Para este modelo inicial de aprovisionamiento, las FA aquí descritas están encerradas dentro de los procesos de *operaciones y gestión del servicio*, y *operaciones y gestión de los recursos*, según el modelo eTOM [23]. La gestión para la relación con el *cliente* y el *proveedor*, CRM y S/PRM, respectivamente, necesita de un modelo de aprovisionamiento más riguroso, compacto, y alineado con la dinámica del operador frente a la oferta y demanda de servicios convergentes.

A.4 Consideraciones de contexto para servicios convergentes

Teniendo en cuenta la dinámica de los servicios convergentes, donde el usuario tiene un rol relevante en la construcción, control y gestión de los mismos, se hace necesaria la existencia de procesos que determinen un contexto de información específico para el cumplimiento de los requisitos del usuario, generando en este un alto porcentaje de aceptación.

Actualmente el usuario final tiene un rol significativo en el aprovisionamiento de servicios, en gran parte porque este se convierte en el punto central de los modelos de negocio. Cada vez son más los usuarios que diseñan estrategias de negocio atractivas a través de las redes sociales, implicando una demanda de servicios y contenido de esa naturaleza. En consecuencia, los proveedores de aplicaciones, de contenido y desarrolladores de servicios de telecomunicaciones posicionan al usuario final como el elemento central en la cadena de aprovisionamiento, generando impactos significativos en los modelos de negocio tanto para aquellos que proveen en el servicio como para aquellos que generan ingresos usándolos.

Teniendo en cuenta este panorama, el aprovisionamiento de servicios convergentes deberá incluir procesos de contexto de información como:

- *Contexto de usuario*: proceso responsable de recolección de la información de usuario como sus preferencias y ubicación.
- *Contexto de dispositivo*: proceso encargado de recolectar la información de los componentes software y hardware del dispositivo del usuario.
- *Contexto social usuario*: proceso responsable de recolectar la información de tipo social del usuario.

- *Contexto de seguridad y privacidad:* proceso encargado de proporcionar y cumplir con las políticas de seguridad de la información de usuario y de aquella suministrada por este.

Finalmente, el modelo inicial de aprovisionamiento de servicios convergentes aprovecha los beneficios del intercambio de mensajes y comunicación REST/JSON, dejando de lado el consumo excesivo de recursos a través de SOAP/XML.

2. RESULTADOS Y DISCUSIÓN

El objetivo de definir un modelo inicial para el aprovisionamiento de servicios convergentes es la identificación de procesos genéricos en cada una de las fases que lo componen. Esta identificación permite la construcción de mecanismos, técnicas y herramientas independientes que agilicen y optimicen dichos procesos. En ese sentido, la figura 3 presenta una posible implementación para cada una de las fases del modelo, especialmente, se describe un mecanismo que incluye de manera automática los procesos de *selección, configuración, activación e instalación* para la fase de despliegue del modelo de aprovisionamiento (etapa de despliegue inicial). Finalmente, se presentan los resultados de la evaluación del prototipo con base en la prueba de escalabilidad.



Figura 3. Implementación fases de aprovisionamiento para servicios convergentes

Fuente: Propia

Los *entornos de creación de servicios* (SCE, Service Creation Environment) facilitan la ejecución de las etapas de diseño, construcción y validación. No obstante, la mayoría de aproximaciones existentes están enfocadas en ambientes asistidos y dependientes del conocimiento de desarrollador. Un mayor aporte a modelos centrados al usuario final recae en el uso de aproximaciones automáticas haciendo uso de tecnologías de lenguaje natural [24].

Los *sistemas de operación del servicio* se encargan de controlar y administrar los recursos para que el usuario consuma el servicio. Diferentes sistemas han empleado tecnologías robustas para soportar los procesos de esta fase, en particular, IBM Oracle y Ericsson se destacan en este mercado.

El *sistema de reconocimiento de contexto* integra aquellos procesos para identificar características en un ambiente convergente. Para ello, soluciones técnicas empleadas por tecnologías como oneAPI [25] pueden emplearse para el acceso de información de contexto de usuario, de contexto de dispositivo, de contexto de servicio, y de actividades de seguridad y privacidad, haciendo uso de la infraestructura de un operador de telecomunicaciones. Adicionalmente, API de Web 2.0 para un contexto social facilita la obtención de perfil de Facebook, Tuenti, twitter, google social del usuario.

El *entorno de ejecución de la lógica del servicio* (SLEE) ofrece las capacidades para ejecutar las funciones implementadas en los servicios que en conjunto trabajan para cumplir con una meta de negocio; además, brinda funciones comunes para realizar el control y gestión del ciclo de vida de los servicios.

El presente trabajo hace uso de un ambiente de ejecución como JAIN Service Logic Execution Environment (JSLEE) para el despliegue automático de servicios convergentes, con base en los procesos de selección, activación, configuración e instalación (etapa despliegue inicial).

JSLEE, a diferencia de otras tecnologías como SIP Servlet o IMS SCIM (SCIM, Service Capability Interaction Manager), se caracteriza por su fácil crecimiento, alta disponibilidad, gestión para el control del estado de los servicios, y por ser un entorno de ejecución lo suficientemente robusto y con altos niveles de desempeño [26]; todos ellos, factores críticos para un operador de telecomunicaciones. Adicionalmente, es una especificación diseñada y optimizada para soportar aplicaciones conducidas por eventos, también conocidas como aplicaciones asíncronas [27]; sus principales elementos son el *modelo de componentes*, el *adaptador de recursos*, el *enrutador de eventos* y las *facilidades de gestión*. El primero es el elemento esencial de la arquitectura del SLEE y está estructurado por bloques de construcción (SBB, Service Building Blocks), donde se describe la lógica funcional del servicio; el segundo tiende la comunicación con entidades externas al SLEE a través de la adaptación de diferentes protocolos (*REST, SOAP, HTTP, Diameter, entre otros*); el tercero enruta los eventos producidos en el interior o en el exterior del SLEE hacia el componente lógico o SBB específico. Finalmente, las facilidades de gestión facilitan el control y monitoreo de los SBB implementados al interior del SLEE [27].

A. Mecanismo de despliegue automático

El mecanismo de despliegue implementado ejecuta de manera automática los procesos de selección, configuración, activación e instalación de los componentes que forman parte del servicio convergente. Para ello, se diseñó una estructura de despliegue compuesta por los módulos: *compilador*, *configuración DU* y *Carga del servicio*, los

cuales analizan, verifican, compilan y adaptan los componentes del servicio al entorno convergente JSLEE. La figura 4 presenta los módulos que hacen parte del mecanismo de despliegue.

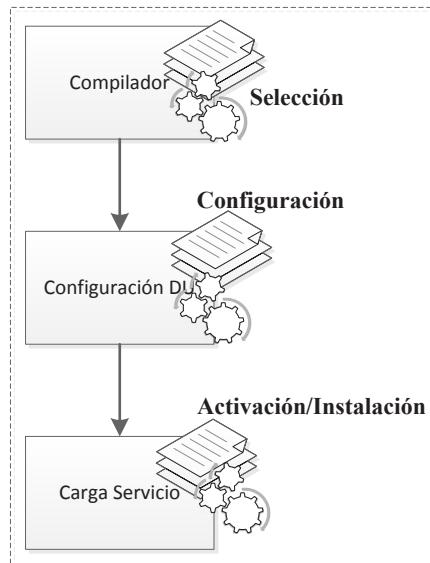


Figura 4. Mecanismo Despliegue Automático

Fuente: Propia

Compilador: este componente tiene como función revisar la sintaxis del código java y la estructura del servicio de acuerdo con la estructura definida por la tecnología JAINSLEE. El producto final es un *.class* que contiene la lógica funcional del servicio convergente. Este módulo implementa el proceso de selección.

Configuración DU: este módulo es el encargado de definir la estructura de despliegue del servicio convergente, en particular, la creación de la Unidad Desplegable (DU, Deployable Unit) y sus archivos de configuración. Implementa el proceso de configuración.

Carga: módulo encargado de la carga del servicio en el servidor para su posterior ejecución. Para este prototipo la carga del servicio es estática, lo cual implica que la DU del servicio permanece fija al interior del SLEE, implicando el consumo de recursos y capacidades del entorno de ejecución. Para modificar esto, debe implementarse el módulo *etapa gestión despliegue* para realizar una gestión eficiente del ciclo de vida del servicio.

B. Validación del mecanismo de despliegue automático

La validación del mecanismo automático de despliegue de servicios convergentes se centra en el consumo del recurso tiempo como métrica de desempeño del sistema. Trabajos como [28, 29] emplean esta métrica como medida de eficiencia para el proceso de ejecución de servicios en plataformas JSLEE. Para este trabajo se diseñó el experimento de escalabilidad, donde se aumenta la complejidad de la solicitud enviada [30].

B.1 Escenario de prueba

Teniendo en cuenta las ventajas de la arquitectura orientada a servicios (SOA, Service Oriented Architecture), se empleó un servicio web contenedor del algoritmo de despliegue automático para servicios convergentes. En ese sentido, se hizo uso de la herramienta SOAPUI para la configuración de los parámetros de la prueba. Esta herramienta es de código abierto y distribuida bajo licencia tipo GNU LGPL; además, facilita la rápida creación de test avanzados de desempeño y evaluación de servicios web [31].

La figura 5 ilustra el escenario de la prueba.

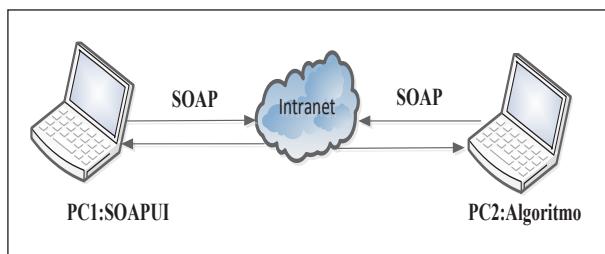


Figura 5. Escenario de prueba de desempeño.

Fuente: Propia

La prueba se realizó haciendo uso de dos PC, de la herramienta SOAPUI y de una conexión de red entre los equipos contendores del algoritmo y de la herramienta de simulación.

El PC1 contiene la herramienta SOAPUI y corre sobre un sistema Windows core i5 de 4GB de RAM; el PC2 contiene el algoritmo de despliegue automático y corre sobre un sistema Linux Ubuntu 12.04, core i5 y 4GB de RAM; la Intranet realiza la conexión entre los PC para la comunicación entre la herramienta de simulación y el algoritmo de despliegue.

El experimento de escalabilidad hace referencia la complejidad de la solicitud [30]. En ese sentido, se enviaron n servicios básicos (Web o Telco), variando n de dos (2) en dos (2). Los parámetros de configuración fueron los siguientes:

- Número de hilos: 1
- Total de ejecuciones: 7
- Tiempo entre ejecuciones: 3 segundos
- Número de servicios básicos (Web oTelco): n , donde $n = \{2, 4, 6, \dots, 30\}$

El *número de hilos* hace referencia a un solo servicio convergente con n servicios básicos; los servicios básicos van aumentando de dos en dos hasta un límite de 30 servicios dentro de la estructura del servicio convergente.

El *Total de ejecuciones* hace referencia al número de veces que se realiza la misma prueba. Esto se hace con el objetivo de encontrar una medida más precisa del tiempo empleado. Para ello, se calcula la media geométrica del conjunto de medidas obtenidas, lo que da como resultado un valor medio más significativo.

Tiempo entre ejecuciones: hace referencia al tiempo de espera entre cada una de las veces que se realiza la prueba. Para este caso se ha dispuesto un tiempo de 3 segundos, suficiente para evitar traslapos entre solicitudes.

Número de servicios básicos (Web o Telco): hace referencia a la cantidad de servicios que componen el servicio convergente. Para cada prueba se aumentan estos servicios, variando la cantidad de 2 en 2 hasta un límite de 30.

La prueba de desempeño cuenta con 10 pruebas de escalabilidad. Para cada una de ellas el total de ejecuciones, el número de hilos y el tiempo entre ejecuciones se mantiene; el número de servicios básicos varía con base en el intervalo de incremento definido.

La figura 6 presenta los resultados obtenidos de la prueba de desempeño.

Como muestra la figura 6, el tiempo total que toma el proceso automático de despliegue de servicios convergentes no supera los 50 ms, lo cual implica un uso eficiente de recursos computacionales del algoritmo. Este tiempo de procesamiento está representado principalmente por el proceso de *selección* (compilador) de los servicios básicos y de elementos de comunicación que componen la lógica del servicio convergente; este proceso hace uso de una cantidad mayor de recursos, principalmente, la generación del archivo *.class* adaptado a la tecnología JSLEE. En efecto, este proceso verifica que el código que representa el servicio convergente esté alineado con la estructura de entorno de ejecución.

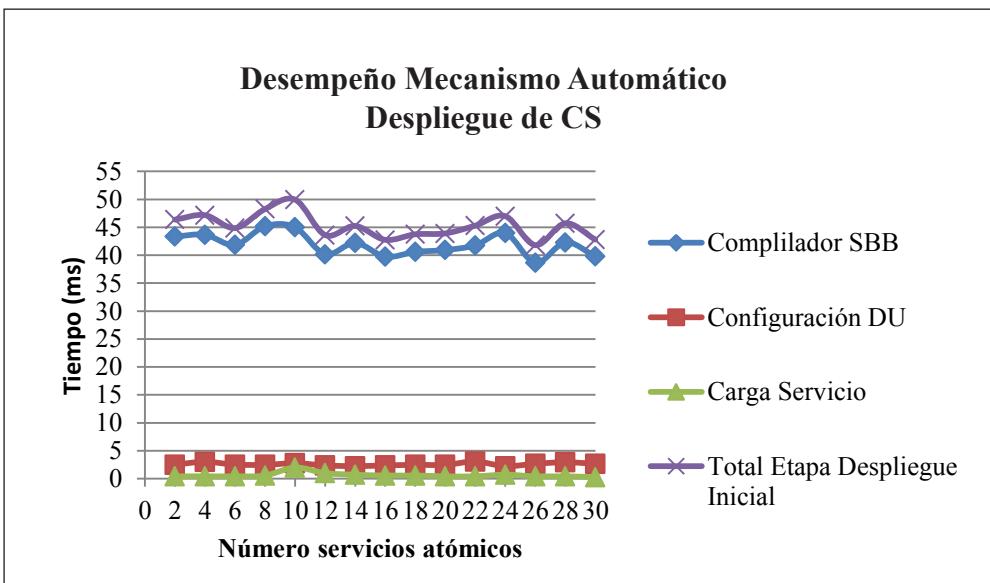


Figura 6. Desempeño mecanismo despliegue automático de CS

Fuente: Propia

De igual manera, el uso de un archivo esqueleto con base en el proceso de *configuración, activación e instalación* para el despliegue en la tecnología JSLEE facilita la configuración y carga de la DU del servicio; en consecuencia, el tiempo empleado por estas fases es mínimo, mejorando con ello del desempeño del sistema. Adicionalmente, el entorno convergente JSLEE es óptimo para la gestión del despliegue del servicio, e implica un consumo mínimo de los recursos computacionales.

Por otro lado, en el inicio de la prueba se obtiene un máximo, esto debido a que el sistema está configurando los archivos necesarios para procesar la solicitud; una vez el sistema supera la etapa de calentamiento, los tiempos empleados por el mecanismo automático de despliegue con base en los procesos de selección, configuración, activación e instalación no superan el máximo presentado. En efecto, procesos de *configuración DU* y *carga del servicio* no proyectan tiempos elevados para servicios compuestos por servicios básicos mayores de 30, facilitando la migración de este mecanismo a un entorno real de aprovisionamiento de servicios convergentes.

Finalmente, es posible concluir que el mecanismo automático de despliegue hace uso eficiente de los recursos computacionales, específicamente, en el tiempo empleado en ejecutar el proceso de despliegue de servicios convergentes para los procesos de selección, configuración, activación e instalación.

3. CONCLUSIONES

Se propuso un modelo inicial de aprovisionamiento de servicios convergentes haciendo énfasis en el ciclo de vida del servicio, el cual sirvió como referente para la construcción de un mecanismo automático de despliegue inicial, enfocándose en el consumo eficiente del tiempo como métrica de desempeño. A continuación se destacan los principales aportes de la investigación:

- El modelo inicial propuesto facilita la creación de mecanismos automáticos de ejecución de actividades claves para el aprovisionamiento de servicios convergentes, principalmente por la modularización y representación genérica de los procesos contenidos en cada una de las fases definidas.
- Las consideraciones de contexto dentro del modelo de aprovisionamiento de servicios convergentes permiten una sinergia con el usuario final, considerándolo como punto central en el modelo de negocio del operador Telco o de un proveedor de aplicaciones, especialmente, el contexto social potencia un modelo de negocio altamente competitivo.
- Los procesos especificados en el modelo inicial de aprovisionamiento reducirán de forma significativa el tiempo de creación, despliegue y operación de servicios avanzados de telecomunicaciones que integren funcionalidades de la Web2.0
- La tecnología JSLEE a través de sus características de reúso de componentes, de las facilidades de gestión, del paradigma asíncrono y del adaptador de recursos, permiten una rápida adaptación y facilidad de crecimiento en un entorno de aprovisionamiento de servicios convergentes.
- El mecanismo automático implementado con base en los procesos definidos en la fase de despliegue inicial del modelo de aprovisionamiento de servicios convergentes, mostró un resultado óptimo en relación con el uso eficiente del tiempo empleado por dicho proceso.
- El mecanismo automático diseñado agiliza y optimiza el proceso de despliegue inicial, gracias a la implementación de componente modulares como el *compilador SBB*, *configuración DU* y *carga del servicio*, los cuales ejecutan las actividades genéricas de selección, configuración, activación e instalación del servicio convergente.
- Teniendo en cuenta que en la práctica diseñar un servicio compuesto con gran cantidad de servicios básicos es una tarea compleja de realizar, principalmente, por la dificultad en el control de las interacciones de las entidades que forman

parte del modelo de negocio del servicio, la prueba de escalabilidad mostró un uso eficiente del recurso tiempo por parte del mecanismo automático de despliegue inicial de servicios convergente para un máximo de 30 servicios.

4. AGRADECIMIENTOS

Al Departamento Administrativo de Ciencia, Tecnología en Innovación (Colciencias) y a la Universidad del Cauca por financiar este proyecto bajo el programa de Jóvenes Investigadores e Innovadores, y a los miembros del Grupo de Ingeniería Telemática (GIT) que participaron en la realización del proyecto TelComp 2.0.

REFERENCIAS

- [1] Devoteam Group, «Service Delivery Platforms: the key to service convergence», Levallois-Perret, 2007.
- [2] J. Zuidweg, «Middleware en Telecomunicaciones», 16 marzo 2009. [En línea]. Available: <http://www.tecsidel.es/tecsidel/index.php?id=896&L=2>.
- [3] A. Sánchez, C. Baladrón, J. Aguiar, B. Carro, L.-W. Goix, J. Sienel, I. Ordás, R. Trapero y A. Bascuñana, «User-centric Service Creation and Execution», de *At Your Service, Service-Oriented Computing From An Eu Perspective*, Londres, MIT Press, 2009, pp. 273-298.
- [4] M. Kundan, OSS for Telecom Networks: An Introduction to Network Management, Springer, 2004.
- [5] O. Chudnovskyy, F. Weinhold, H. Gebhardt y M. Gaedke, «Integration of Telco Services into Enterprise Mashup Applications», de *Proceedings of ComposableWeb-Springer*, Chemnitz, Germany, 2011.
- [6] J. C. Yelmo, J. M. del Álamo, R. Trapero y Y.-S. Martín, «A user-centric approach to service creation and delivery over next generation networks», *Computer Communications*, vol. 34, pp. 209-222, Abril 2011.
- [7] Yahoo, «Pipes,» 2012. [En línea]. Available: <http://pipes.yahoo.com/pipes/>. [Último acceso: 15 enero 2012].
- [8] A. V. Riabov, E. Bouillet, M. D. Feblowitz, Z. Liu y A. Ranganathan, «Wishful Search: Interactive Composition of Data Mashups», de *17th International World Wide Web Conference*, Beijin, 2008.
- [9] A. Hristoskova, B. Volckaert, F. De Turck y B. Dhoedt, «Design of a Framework for Automated Service Mashup Creation and Execution», 2010.
- [10] E. Gonçalves da Silva, L. Ferreira Pires y M. van Sinderen, «Towards runtime discovery, selection and composition of semantic services», *Computer Communication*, vol. 34, pp. 159-168, abril 2011.

- [11] K. S. C. Toh y M. S. M. F. D. Syed, «Towards Building Semantic Next Generation Network – A Preliminary Study», *Procedia Computer Science*, vol. 5, pp. 689-696, 2011.
- [12] Xiuquan Qiao, Xiaofeng Li y Junliang Chen, «Telecommunications Service Domain Ontology: Semantic Interoperation Foundation of Intelligent Integrated Services», *Telecommunications Networks - Current Status and Future Trends*, n.º ISBN: 978-953-51-0341-7, pp. 183-210, marzo 2012.
- [13] C. Bo, Z. Yang, Z. Peng, D. Hua, H. Xiaoxiao, W. Zheng y C. Junliang, «Development of Web-Telecom based hybrid services orchestration and execution middleware over convergence networks», *Journal of Network and Computer Applications*, vol. 33, pp. 620-630, marzo 2010.
- [14] D. Zhu, Y. Zhang, B. Cheng, B. Wu y J. Chen, «HSCEE: A Highly Flexible Environment for Hybrid Service Creation and Execution in Converged Networks», *Journal of Convergence Information Technology*, vol. 6, n.º 3, pp. 264-276, marzo 2011.
- [15] T. Eichelmann, W. Fuhrmann y B. Ghita, «Support of parallel BPEL activities for the Team-Com Service Creation Platform for Next Generation Networks», 2010.
- [16] IBM, «TEL COS- La problemática de los OSS/BSS desintegrados», septiembre 2006. [En línea]. Available: www.gbm.net/bt/bt34/tendencias/telcos.php.
- [17] Oracle, «Oracle», 2012. [En línea]. Available: <http://www.oracle.com/us/industries/communications/overview/index.html>.
- [18] Ericsson, «Ericsson in OSS/BSS», 2012. [En línea]. Available: <http://www.ericsson.com/oss-bss/>.
- [19] Gartner, «Magic Quadrant for Operations Support Systems», octubre 2012. [En línea]. Available: <http://www.gartner.com/technology/reprints.do?id=1-1CMVIU3&ct=121029&st=sb>.
- [20] TeleManagement Forum (TMF), «Service Delivery Framework Reference Architecture», 2009.
- [21] Vanea Chiprianov, «Collaborative Construction of Telecommunications Services. An Enterprise Architecture and Model Driven Engineering Method», Morbihan-Francia, 2012.
- [22] G. Kecskemetia, G. Terstyanszkyb, P. Kacsuka y Z. Nemétha, «An approach for virtual application distribution for service deployment», *Future Generation Computer Systems*, vol. 27, n.º 3, p. 280–289, 2011.
- [23] TMForum, «Business Process Framework (eTOM)», GB921, 2008.
- [24] Armando Ordóñez, Juan C. Corrales, F. Paolo y G. Jaime, «User Centred Automated Composition in Telco 2.0», de *The Fourth International Conference on Advanced Service Computing*, Nice, 2012.
- [25] GSMA, «OneAPI», 2012. [En línea]. Available: <https://oneapi.aepona.com/>.
- [26] OpenCloud, «Rhino 2.1: Overview and Concepts», Wellington, 2009.

- [27] Sun Microsystems, Inc, «JAIN SLEE (JSLEE) 1.1 Specification, Final Release», 2008.
- [28] Femminella, M. , Francescangeli, R., Giacinti, F., Maccherani, E., Parisi, A. y Reali, G., «Scalability and performance evaluation of a JAIN SLEE-based platform for VoIP services», de *Teletraffic Congress, 2009. ITC 21 2009. 21st International*, Paris, 2009.
- [29] G. Rojas Sierra, F. Estrada Solano, J. A. Caicedo M. y O. M. Caicedo R., «Technical Criteria for Value-Added Services Creation, Execution and Deployment, on Next Generation Networks», de *7th Advanced International Conference on Telecommunications (AICT)*, St. Maarten, 2011.
- [30] C. Kankanamge, Web Services Testing with soapUI, Packt Publishing Ltd, 2012.
- [31] S. Hussain, Z. Wang, I. Kalil y A. Diop, «Web Service Testing Tools: A Comparative Study», de *IJCSI International Journal of Computer Science*, 2013.