

La Plataforma Telcoblocks de Despliegue y Desarrollo de Servicios VoIP

Jonathan González

Depto Ingeniería de Sistemas Telemáticos
Universidad Politécnica de Madrid
Ciudad Universitaria s/n 28040 Madrid
jonathan.gsanchez@alumnos.upm.es

Carlos A. Iglesias

División de I+D+i
Germinus XXI (Grupo Gesfor)
Avda Manoteras, 32 28050 Madrid
cif@germinus.com

Felipe Echanique

Depto Ingeniería de Sistemas Telemáticos
Universidad Politécnica de Madrid
Ciudad Universitaria s/n 28040 Madrid
felipe.echanique.torres@alumnos.upm.es

Resumen—Este artículo presenta el entorno de desarrollo y despliegue de servicios VoIP propuesto dentro del proyecto TelcoBlocks. En concreto, se detalla el componente de personalización propuesto que facilita la personalización de servicios construidos con tecnologías JAIN SLEE o SIP Servlets. El componente ha sido aplicado al desarrollo de un servicio de personalización de tonos, así como a un servicio de personalización de anuncios en un servicio de Click to Dial.

Palabras Clave—Telcoblocks, personalización, VoIP, Java, SIP, SIP Servlets, JAIN SLEE

I. INTRODUCCIÓN

La convergencia de las redes de telefonía tradicional e Internet [21] está proporcionando una nueva arquitectura para el desarrollo de servicios telco en las así llamadas Redes de Nueva Generación (*Next Generation Networks*; NGN [3]). Una de las principales características de NGNs [2] es el desacoplamiento de redes y servicios, permitiendo que se ofrezca de forma separada y que evolucionen independientemente.

Los principales retos para esta nueva arquitectura NGN son por una parte, proporcionar un entorno de ejecución de servicios tolerante a fallos y con calidad de operadora (*carrier grade*), que sea al menos tan robusto y seguro como la actual Red de Telefonía Conmutada (RTC); por otra parte, debe ofrecer un entorno flexible y abierto que permita desarrollar nuevos servicios de forma ágil, dado que la alta competitividad del entorno telco está demandando esta flexibilidad.

Para conseguir esta flexibilidad, Moyer [21] señala que NGN debe ser más abierta que RTC a los desarrolladores de servicios, ofreciendo APIs abiertas y facilitando bloques de servicios primitivos que los desarrolladores puedan reutilizar. En esta línea, el software de telecomunicaciones NGN ha seguido tres enfoques: JAIN SLEE [16], Parlay [7] y SIP Servlets [22].

El desarrollo de aplicaciones web 2.0 complementa [13] los servidores de aplicaciones híbridos JavaEE / SIP ofrecidos en los modelos de arquitectura orientados a servicios, y cuestiona si la capa de servicios IMS debe seguir siendo una capa separada, orientada a aplicaciones telco, o integrada con las aplicaciones web 2.0. Esta convergencia de servicios telco y web se está comenzando a denominar *mashups telco web2.0* [13]. En esta misma línea, la tecnología web de mashups está comenzando a usarse para la construcción de servicios telco por parte de los usuarios [9].

A pesar de disponer de estas facilidades, el desarrollo de aplicaciones en entornos telco aún no se ha popularizado,

debido a la complejidad de las tecnologías involucradas, así como a la falta de entornos que faciliten su adopción. El objetivo del proyecto Telcoblocks [8] es el desarrollo de una plataforma abierta que facilite el desarrollo, despliegue, prueba e integración de servicios VoIP, tanto en operadoras como en empresas que disponen de una centralita VoIP. Telcoblocks es un proyecto orientado a los desarrolladores, con el fin de reducir drásticamente los tiempos de desarrollo y despliegue de nuevos servicios. Uno de los pilares del proyecto es la reutilización de componentes y servicios para la construcción de nuevos servicios.

El resto del artículo se estructura como sigue. La sección II presenta el contexto de esta investigación, que se enmarca en el proyecto TelcoBlocks, detallando la plataforma de ejecución y desarrollo de servicios definida. A continuación, la sección III detalla el componente de personalización, objeto principal de este artículo, que facilita la integración de funcionalidades de personalización en la construcción de un servicio (anuncios, facturación, tarificación, etc.). La sección IV presenta un caso de estudio del componente de personalización presentado en la sección III y desplegado en la infraestructura presentada en la sección II. Por último, se presentan conclusiones del trabajo realizado y las líneas actuales de trabajo en la sección V.

II. TELCOBLOCKS: UN ENTORNO PARA DESARROLLO, EJECUCIÓN Y PRUEBA DE SERVICIOS VOIP

Telcoblocks nace con el objetivo de proveer una plataforma abierta para el desarrollo y ejecución de servicios de telecomunicación para la Red de Nueva Generación (NGN[3]) empleando tecnologías JAIN SLEE [16], SIP Servlets [22] y Parlay [7].

El proyecto investiga en el desarrollo de herramientas de software libre para entornos telco que cumplan con los fuertes requisitos de disponibilidad y fiabilidad de estas redes. Como resultado, la plataforma resultante se liberará como código abierto con licencia GPL, lo cual supone un cambio de mentalidad frente el uso exclusivo de software propietario en este tipo de entornos.

El proyecto también innova en la aplicación de sistemas inteligentes a las telecomunicaciones. Frente el fenómeno emergente de la sobre-comunicación potenciado por el auge de las comunicaciones personales, el proyecto investiga en la integración de técnicas inteligentes para gestionar dichas comunicaciones según las preferencias de los usuarios, así como para facilitar la personalización de servicios.

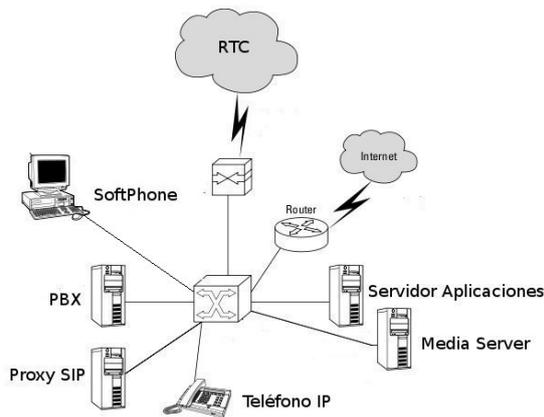


Figura 1. Plataforma de Despliegue de Servicios VoIP de Telcoblocks

Telcoblocks enmarca los aspectos mencionados dentro de procesos de desarrollo ágil de servicios telco basados en tecnologías abiertas y estándares, cuyo ahorro en costes y menor time-to-market abre interesantes oportunidades de negocio a las PYMES para proporcionar servicios de telecomunicación a usuarios finales.

Telcoblocks desarrolla dos plataformas: la plataforma para despliegue de servicios (sección II-A) y la plataforma de desarrollo de servicios (sección II-B).

II-A. Plataforma de Despliegue de Servicios VoIP

En los últimos años, la gran aceptación que ha tenido la VoIP como solución a las comunicaciones corporativas de las empresas ha favorecido al nacimiento de algunos proyectos de software abierto que permiten implementar la mayoría de las características que ofrecen algunas de las PBX comerciales (Avaya, Cisco...) que desde el nacimiento de esta tecnología han sido utilizadas por las grandes y medianas empresas.

En la actualidad existen diferentes soluciones de software abierto para el despliegue de una infraestructura VoIP basada en el protocolo SIP [18]. TelcoBlocks pretende investigar en la integración de estas soluciones para crear una arquitectura integrada que permita generar un entorno de pruebas y despliegue de servicios de forma sencilla utilizando diferentes elementos.

El entorno básico de despliegue de servicios VoIP de Telcoblocks se ilustra en la figura 1 y consta de los siguientes elementos, que serán descritos en la siguiente sección:

- Centralita VoIP (PBX) (sección II-A1).
- Proxy SIP - Servidor de Registro (sección II-A2).
- Servidor Multimedia (sección II-A3).
- Servidor Telco de Aplicaciones (sección II-A4).

II-A1. Centralita VoIP (PBX): Una PBX (*Private Branch Exchange*) es una central telefónica conectada directamente a la red pública de teléfono por medio de líneas troncales para gestionar, además de las llamadas internas, las entrantes y/o salientes con autonomía sobre cualquier otra central telefónica. Los usuarios de una PBX no tienen asociada ninguna central de teléfono pública, ya que es el mismo PBX

que actúa como tal, análogo a una central pública que da cobertura a todo un sector mientras que un PBX lo ofrece a las instalaciones de una compañía generalmente.

Una centralita VoIP (PBX) ofrece:

- Salida a la Red Telefónica Conmutada (RTC): Los usuarios registrados en el sistema podrán realizar llamadas a la RTC.
- Llamadas a bajo coste: Una PBX es capaz de conectarse a diferentes proveedores de servicios de VoIP, permitiendo así la realización de llamadas que serán tarifadas a la plataforma por el proveedor con el que se realice la llamada.

La solución de código abierto más empleada en el ámbito profesional es Asterisk [5]. Asterisk es una aplicación de software libre que implementa las funciones de una central telefónica (PBX). La aplicación es compatible con diversos sistemas operativos (Linux, MacOS, Solaris y Microsoft Windows) aunque es mejor soportada en sistemas Linux.

Como toda PBX, Asterisk permite la interconexión de diferentes teléfonos que pueden ser registrados en la aplicación y ser localizados entre sí mediante la marcación de diferentes extensiones definidas en un plan de llamadas. Además, permite conectarse con proveedores de llamadas VoIP de forma que un teléfono registrado en Asterisk es capaz de realizar llamadas a números de la RTC a bajo precio.

Una de las características más importantes de Asterisk es el hecho de que puede ser usado con diferentes protocolos de señalización de VoIP; entre ellos destacar H.323 y SIP aunque también es muy utilizado el protocolo IAX (propietario de Asterisk) [19] para comunicación entre diferentes PBX de este tipo.

El hecho de ser una aplicación de software libre la hace realmente atractiva, debido a las mejoras que se van añadiendo en las nuevas versiones que son publicadas cada poco tiempo. Actualmente se encuentra publicada la versión 1.4.x de Asterisk a la espera de que la versión de 1.6.x sea publicada en los próximos meses.

Un proyecto que ha sido analizado por su sencillez de configuración es Elastix [27] que incluye junto con un Asterisk configurable desde una interfaz web (FreePBX) un sistema de facturación muy sencillo de configurar.

II-A2. Proxy SIP - Servidor de Registro:

Cada usuario tiene una dirección lógica que es invariable respecto de la ubicación física del usuario. Una dirección lógica del protocolo SIP es de la forma usuario@dominio es decir tiene la misma forma que una dirección de correo electrónico. La dirección física (denominada "dirección de contacto") es dependiente del lugar en donde el usuario está conectado (de su dirección IP).

Cuando un usuario inicializa su terminal (por ejemplo conectando su teléfono o softphone SIP), el agente de usuario SIP que reside en dicho terminal envía una petición con el método REGISTER a un Servidor de Registro, informando a qué dirección física debe asociarse la dirección lógica del usuario. El servidor de registro realiza entonces dicha asociación. Esta asociación tiene un período de vigencia y si no es renovada, caduca. También puede terminarse mediante un desregistro. La forma en que dicha asociación es almacenada en la red no es determinada por el protocolo SIP, pero es vital

que los elementos de la red SIP accedan a dicha información.

Además, en el caso de la arquitectura propuesta, este servidor actúa también como servidor proxy o de redirección. Para encaminar un mensaje entre un agente de usuario cliente y un agente de usuario servidor normalmente se recurre a estos servidores. Estos servidores pueden actuar de dos maneras:

1. Como Proxy, encaminando el mensaje hacia destino.
2. Como servidor de Redirección generando una respuesta que indica al originante la dirección del destino o de otro servidor que lo acerque al destino. La principal diferencia es que el servidor proxy queda formando parte del camino entre los extremos de la comunicación, mientras que el servidor de redirección una vez que indica al llamante cómo encaminar el mensaje ya no interviene más.

En el marco del proyecto Telcoblocks, se han estudiado las posibles soluciones de código abierto que se podían emplear en esta arquitectura entre las que destacan: SER (SIP Express Router) [4], OpenSER [6] (OpenSIPS, Kamailio). Tras un análisis de cada uno de los proyectos se optó por el uso de OpenSIPS, ya que parte de la última versión de OpenSER) y es de las soluciones más utilizadas en el ámbito profesional junto con Asterisk.

OpenSIPS es una aplicación que implementa un servidor SIP. Originalmente pertenecía a otro proyecto que sigue estando activo llamado SER (*Sip Express Router*). La principal diferencia entre ambos proyectos es el hecho de que OpenSER está llevado por una comunidad en lugar de una empresa, lo que hace que rápidamente se implementen mejoras en el servidor y las empresas se inclinen por la opción de uso de OpenSER. OpenSIPS es muy versátil para su implantación, permitiendo su instalación en sistemas con recursos limitados así como en grandes servidores. Está escrito completamente en C y orientado principalmente a equipos Linux/Unix. Esta aplicación tiene muchas características interesantes, entre las que podemos destacar las siguientes: Location Service, registrar, servidor Proxy, servidor de redirección de llamadas, gateway hacia otras redes que no son SIP. Al igual que Asterisk, OpenSIPS permite la interconexión con diferentes terminales y llamadas a través de la RTC.

II-A3. Servidor Multimedia: Un Servidor Multimedia (*Media Server*) permite la implementación de muchos de los servicios mencionados en la descripción de la centralita VoIP, la diferencia fundamental es que no registra usuarios sino que solo se encarga del tráfico RTP.

De entre las soluciones que implementan un servidor multimedia, han sido consideradas la solución de Mobicents (*Mobicents Media Server* [28]) y la solución del proyecto SER, llamada SEMS [26]. Por facilidad de integración con el servidor SIP escogido se ha optado por esta última opción.

SEMS es un servidor de aplicaciones y recursos multimedia para servicios VoIP basados en SIP. Presenta muy buenas prestaciones realizando algunos servicios básicos como anuncios, conferencia combinándose, en algunos casos, con servidores de aplicaciones externos. Gracias a su facilidad de uso y su framework de aplicaciones flexible, así como de su soporte Agentes de Usuario extremo a extremo, se puede unir en un mismo proceso la lógica de la aplicación con los recursos servidos por el servidor. Suele ser empleado con algunos de

los servidores SIP analizados anteriormente (SER, OpenSER, OpenSIPS) de tal manera que se obtiene así un servicio de VoIP completo. Las principales funcionalidades de SEMS son las siguientes:

- Servicio de Conferencia: Permite conversaciones telefónicas por más de dos usuarios simultáneamente.
- Servicio de Videoconferencia.
- Anuncio: Reproduce un recurso solicitado.
- Servicio de Voicemail: Servicio que almacena mensajes destinados a usuarios que en el momento de la recepción de los mismos no pudieron atender la llamada. Estos mensajes son enviados posteriormente al correo electrónico del usuario.

II-A4. Servidor Telco de Aplicaciones: Ejecuta la lógica de negocio del servicio (ClickToDial, tarificación de llamadas, personalización de servicios) Telcoblocks actualmente soporta dos tecnologías: JAIN SLEE [16] y SIP Servlets [22]. Como servidor JAIN SLEE, se ha escogido Mobicents [24], que soporta JAIN SLEE y SIP Servlets. Como servidor de SIP Servlets se ha escogido Sailfin [1], que soporta SIP Servlets, gracias a sus facilidades de desarrollo integradas con el IDE Netbeans.

Una vez analizadas cada una de las soluciones escogidas para la arquitectura se puede concluir afirmando que las principales ventajas de esta arquitectura son:

1. Escalabilidad de servicios, gracias al uso de un Servidor Telco de aplicaciones
2. Soporte al protocolo AAA [25]: Integración Radius [12], Diameter [23], LDAP [20], gracias a la integración de OpenSIPS.
3. Facturación y conectividad a RTC ofrecidas por Asterisk.

La plataforma de despliegue se ha empaquetado como una **máquina virtual** con VmWare [29], para que los desarrolladores puedan disponer de un entorno donde puedan probar rápidamente los servicios desarrollados. Dicha máquina virtual, contiene todos los elementos mencionados en la arquitectura, adecuadamente configurados, para que el desarrollador sólo deba estar pendiente del nuevo servicio que esté implementando.

II-B. Plataforma de Desarrollo de Servicios VoIP

Actualmente, el desarrollo de servicios telco manifiesta ciertas deficiencias, como son la baja productividad, el elevado tiempo dedicado al desarrollo así como las altas habilidades que necesitan los desarrolladores para conocer e integrar los actuales frameworks y componentes.

Para resolver estas deficiencias, Telcoblocks pretende abordar dos direcciones: mejorar la prueba de los servicios, y reducir el tiempo de desarrollo mediante la reutilización de componentes, que redundarán en una mejora de su calidad.

La plataforma de desarrollo requiere que se facilite el desarrollo con dos tecnologías, JAIN SLEE y SIP Servlets, que se presentan a continuación brevemente.

II-B1. JAIN SLEE: JAIN SLEE [16] es un modelo de componentes definido para un servidor de aplicaciones diseñado específicamente para aplicaciones telco. Está concebido como una plataforma de procesamiento de eventos de altas prestaciones y tolerante a fallos. Frente a las aplicaciones de

empresa y web, síncronas e intensivas en datos por naturaleza, y que pueden ser modeladas e implementadas de forma adecuada con la tecnología JEE (*Java Enterprise Edition*), SLEE está dirigido a aplicaciones asíncronas, como son las aplicaciones telco, y a procesar eventos de red combinando múltiples protocolos. En la actualidad JAIN SLEE es el estándar Java para entornos de ejecución de lógicas de servicio (*Service Logic Execution Environment*) [30].

La principal ventaja de JAIN SLEE es la estandarización del desarrollo y ejecución de servicios, de modo que se cubran los diferentes niveles y, muy especialmente, la capa de acceso a los elementos de telecomunicaciones. A pesar de la estandarización en el acceso a las capacidades de red ofrecida por protocolos como INAP o CAP, los cuales permiten la interconexión de redes, la complejidad de la implementación de servicios de forma directa y los requerimientos de los telcos de implementación (rendimiento, distribución, confiabilidad, etc.) hacen del desarrollo de servicios una actividad muy compleja, que cuenta con tan sólo un reducido grupo de profesionales con la competencia adecuada. JAIN SLEE, por el contrario, ofrece un alto nivel de abstracción para el acceso a las capacidades de red, simplificando en gran medida la complejidad existente y, al mismo tiempo, ofreciendo Java como lenguaje de implementación. Paralelamente, JAIN SLEE incorpora los conceptos tradicionales de la arquitectura Java (reutilización de componentes, facilidades de administración y concurrencia, distribución, etc.).

JAIN SLEE, como paradigma de los entornos de nueva generación, rompe con la tradicional distinción entre servicios de inteligencia de red y servicios IP. El concepto de adaptadores de recursos (RA) permite la integración entre diferentes tecnologías en servicios innovadores y de nueva generación, así como el despliegue de servicios tradicionales basados en señalización, como la gestión de enrutado de llamadas. No existe un único campo de servicios adecuado para JAIN SLEE; por el contrario, una de las principales ventajas es su gran alcance, el cual es además ampliable mediante la incorporación de adaptadores de nuevos protocolos. Otra de las ventajas es su definición abierta, que elimina las barreras tradicionales de integraciones propietarias.

II-B2. SIP Servlets: SIP Servlets [22] proporcionan un modelo de desarrollo específico del protocolo SIP, dado su papel fundamental en las nuevas arquitecturas IMS. Es una tecnología alternativa a JAIN SLEE, que ofrece un modelo de desarrollo más sencillo, aunque no permite manejar otros protocolos diferentes de SIP y no es transaccional.

Telcoblocks va a contemplar una línea de investigación basada en el uso de una biblioteca de componentes, el cual ayudará a los ingenieros software a la selección e integración de bloques para la creación de nuevos servicios.

La plataforma de desarrollo de telcoblocks, está formada por los siguientes elementos:

- *Plugins para Eclipse y Netbeans*, que facilitan el desarrollo de forma gráfica de nuevos servicios. Actualmente se ha implementado un diagrama de despliegue para definir y configurar la plataforma de despliegue de servicios, y un diagrama de bloques de servicio, para combinar módulos desarrollados previamente.
- *Herramientas de prueba y simulación*. El proyecto in-

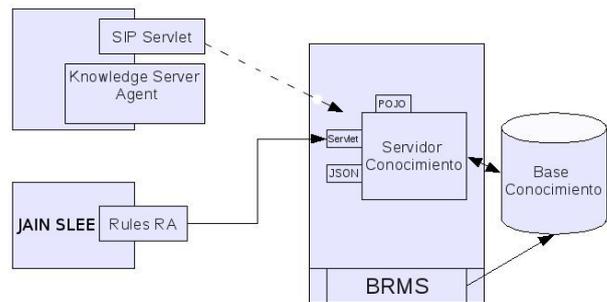


Figura 2. Arquitectura del Componente de Personalización

tegrará el desarrollo de pruebas automáticas así como la integración de pruebas que faciliten el desarrollo de servicios.

- *Biblioteca de componentes*. A través del desarrollo de casos de uso, se han identificado algunos componentes básicos que se reutilizan en numerosos servicios. El primero que se ha implementado es el componente de personalización, que se presenta a continuación.

III. COMPONENTE DE PERSONALIZACIÓN

El objetivo del componente de personalización es facilitar la externalización de la lógica del negocio de un servicio de telecomunicaciones. La personalización [17] es un elemento clave para tanto el descubrimiento de nuevos servicios, como la adaptación de los servicios existentes a las características de los usuarios.

La figura 2 muestra la arquitectura del componente de personalización propuesto, que consta de los siguientes elementos:

- **Servidor de Conocimiento:** es el encargado de gestionar la *base de conocimiento*, ofreciendo una interfaz para su acceso remoto. La base de conocimiento define las reglas y hechos de la lógica de servicio. El servidor está integrado con otro componente, llamado BRMS (*Business Rule Management System*) que permite la edición de la base de conocimiento mediante una interfaz web. Se ha seleccionado la plataforma Drools [11] para su implementación. Ofrece diferentes interfaces para su acceso (clase Java en local, útil para desarrollo, Servlet para acceso remoto, o bien JSON para la capa de presentación). Para el acceso remoto de este servidor de conocimiento, los clientes emplean *RuleAgents*, que son clases Java que hacen de proxy del servidor de conocimiento, encapsulando toda la lógica de acceso.
- **Interfaz para tecnología SIP Servlets.** Los sip servlets atienden las peticiones SIP, haciendo peticiones al servidor de conocimiento cuando de él precisa algo, por ejemplo; en el servicio de anuncio personalizado, realiza una petición al servidor de conocimiento, para saber el anuncio a reproducir, antes de solicitar el anuncio a reproducir al servidor multimedia. Los SIP Servlets usan los componentes *RuleAgent* directamente.
- **Interfaz para tecnología JAIN SLEE.** Para los componentes JAIN SLEE, la integración se ha realizado mediante un adaptador de recursos Rules-RA [10], que

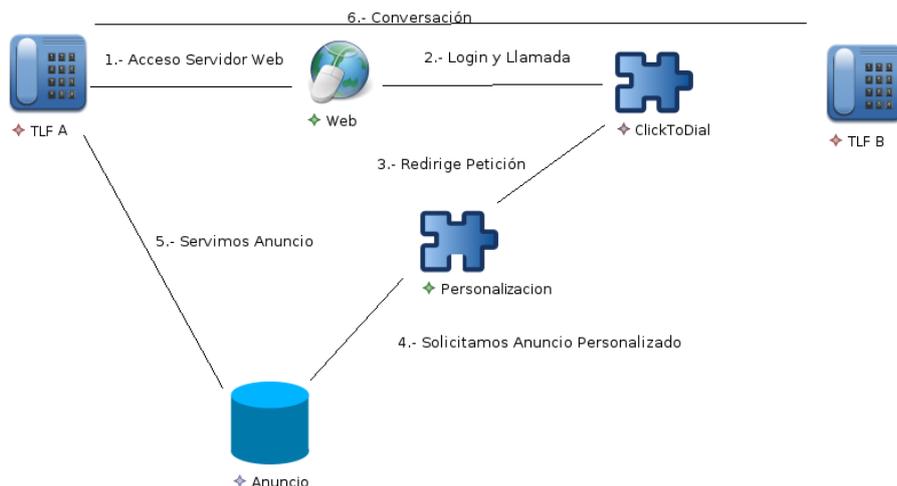


Figura 3. Secuencia Acciones Servicio



Figura 4. Interfaz Web Click To Dial

permite cargar realizar consultas a los componentes SBB. En este caso, es el adaptador de recurso el que emplea un componente *RuleAgent* para acceder al gestor de conocimiento, que es configurable mediante la interfaz de gestión con JMX.

IV. CASO DE ESTUDIO: PERSONALIZACIÓN DE ANUNCIOS PARA EL LLAMANTE

El servicio seleccionado consiste en la personalización de anuncios a un llamante combinado con un servicio de llamadas a través de la web (*clickToDial*, figura 4). Los usuarios se registran en un portal web, especifican una serie de datos en su perfil, y pueden realizar llamadas pinchando en un botón (servicio *ClickToDial*). En el momento en el que se cursa la llamada, el sistema selecciona un anuncio auditivo, que es escuchado por el usuario, y una vez que termina el anuncio se cursa la llamada al destinatario. A cambio de escuchar la llamada, el usuario puede recibir alguna promoción, tales como descuentos o minutos gratis.

La figura 3 muestra el diagrama de servicio desarrollado con el plugin para diagramas de bloques de Telcoblocks para Eclipse, desarrollado con *Graphical Modeling Framework (GMF)* [15]

Primeramente, intentaremos abordar el problema desde el punto de vista de SIP. En el escenario de este servicio debe haber, al menos, dos clientes SIP registrados en nuestro servidor (llamante y destinatario de la llamada). El intercambio completo de mensajes durante la reproducción de anuncio y durante la conversación se muestra en la figura 5.

Una vez registrados ambos terminales, se procede a la realización de la llamada. La petición SIP de llamada (INVITE) es dirigida al registrar, que va a dirigir dicha petición al servidor multimedia (SEMS [26]) que reproducirá el anuncio solicitado. La cabecera de la petición enviada por el registrar al media server tiene un formato especificado en la RFC 4240 [14]. En la cabecera de esta petición debe ir especificado el anuncio que debe ser reproducido por el servidor multimedia, en un parámetro llamado "play=". Una vez finalizada la reproducción del anuncio, se finaliza la sesión para negociar la sesión con el destinatario de la llamada.

Para realizar la selección del anuncio, se emplea el componente de personalización previamente descrito. La base de conocimiento consta de tres inferencias como se muestra en el diagrama de inferencias (figura 6: segmentación de la población, clasificación de anuncios y selección del anuncio). Tal como se indica, actualmente las dos primeras inferencias se realizan en la fase de registro, mientras que la selección del anuncio se realiza en tiempo real con cada llamada.

La segmentación de la población, consiste en clasificar cada uno de los usuarios de la plataforma en cada uno de los segmentos de población que se han definido a partir de criterios de edad y estudio/trabajo. Esta segmentación se realizará cada vez que un usuario se registra en el sistema, o modifica uno de los campos que afecta a la segmentación. No es necesario hacer una segmentación de la población entera cada vez que se modifica un usuario, sino que basta con solo insertar a ese usuario en la base de conocimiento, partiendo del hecho de que los perfiles de usuario son estáticos en este caso.

La clasificación de los anuncios sigue criterios parecidos, clasificándolos en categorías de contenidos y precios cuando los anuncios son dados de alta o modificados. Así pues la clasificación de los anuncios atiende a los siguientes criterios:

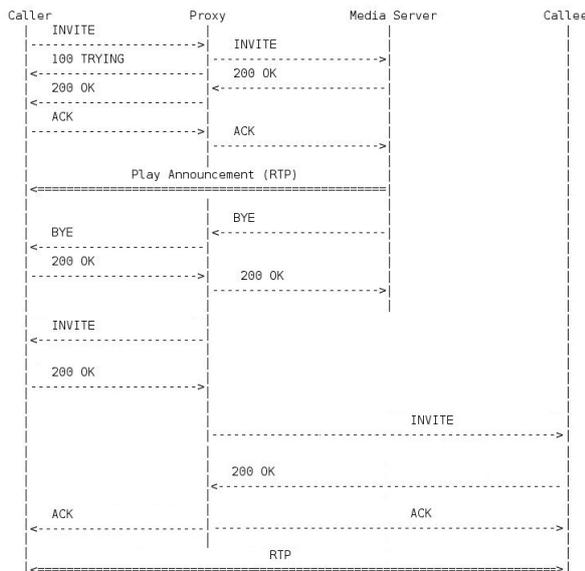


Figura 5. Diagrama de Trazas SIP en el servicio de anuncio personalizado

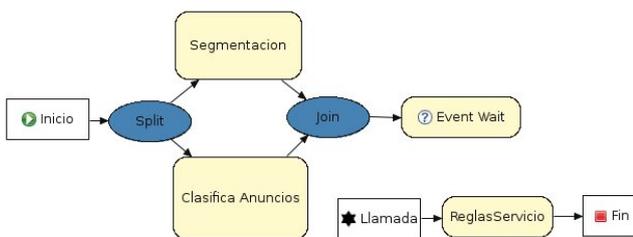


Figura 6. Flujo de Inferencias para la personalización de un anuncio

- Edad, diferenciando si van dirigidos a mayores de edad o no.
- Temática (Ocio, Cultura, Evento...)
- Precio, distinguiendo si va dirigido a un público con un poder adquisitivo elevado o no.

La última fase es la fase de elección del anuncio emplea una técnica de puntuación (*scoring*) de los anuncios según su afinidad al perfil del llamante (por edad, por aficiones, por nivel social (trabajo), región, etc) y a algunas características que tengan en común llamante, usuario llamado y anuncio. Una vez puntuados todos los anuncios siguiendo estos criterios, se elige el de mayor puntuación que será reproducido por el servidor multimedia para posteriormente cursarse la llamada solicitada.

V. CONCLUSIONES Y FUTUROS TRABAJOS

El artículo ha presentado las plataformas de despliegue y desarrollo para servicios VoIP de Telcoblocks, que facilitan y reducen el tiempo de desarrollo de nuevos servicios VoIP.

El componente de personalización facilita su reutilización en varios niveles. Por una parte, facilita la inclusión de personalización en nuevos servicios, reutilizando los com-

ponentes desarrollados y simplemente definiendo una nueva base de conocimiento. Por otra parte, la misma base de conocimiento puede ser usada por varios servicios programados con tecnologías diferentes, gracias al uso de un servidor de conocimiento.

El caso de estudio presentado ha mostrado la facilidad para integrar, por ejemplo, la RFC 4240 en el desarrollo del servicio de anuncios personalizados.

Actualmente, se está trabajando en el uso del componente de personalización en otros ámbitos. Se ha desarrollado su integración con un GoogleTalk mediante componentes JAIN SLEE, lo que muestra la flexibilidad del componente desarrollado.

Otra línea de trabajo actual es el entorno de desarrollo de Telcoblocks, para el que se están desarrollando plugins para Eclipse, realizados con EPF, como se han mostrado en este artículo.

AGRADECIMIENTOS

Este proyecto ha sido cofinanciado por el Ministerio de Industria, Turismo y Comercio mediante el proyecto Telcoblocks (TSI-020302-2008-16) bajo el programa Avanza I+D 2008.

REFERENCIAS

- [1] Sailfin - contenedor de sip servlets. Technical report, sun.
- [2] Conclusions from the ngn-sg. Technical report, ETSI 38th General Assembly meeting Nice, November 2001.
- [3] Definition of next generation network. Technical report, ITU, 2004.
- [4] Sip express router (ser). Technical report, iptel.org, 2004.
- [5] Asterisk the future of telephony. Technical report, asterisk.org, 2005.
- [6] Open sip express router (ser). Technical report, openser.org, 2007.
- [7] Osa parlay x 3.0 specifications. Technical report, ETSI, 2007.
- [8] Proyecto telcoblocks, 2008. Disponible en <http://telcoblocks.germinus.com>.
- [9] A. Alvaro Martínez Reol, C. Baladrón Zorita, A. León Martín, C. García Morchón, L. Calavia Domínguez, J. Aguiar Pérez, and J. Caetano. Nuevos modelos de negocio: Servicios generados por el usuario. In *XVIII Jornadas Telecom I+D, ISBN-13: 978-84-9860-135-0, Octubre 2008, Bilbao*, 2008.
- [10] A. Bhayani. Mobicents rules-ra, disponible en <http://groups.google.com/group/mobicents-public/web/mobicents-rules-ra>, 2007.
- [11] P. Browne. *JBoss Drools Business Rules*. PACKT Publishing, 2009.
- [12] A. R. C. Rigney, S. Willens. Remote authentication dial in user service (radius). Technical report, IETF - RFC 2865, 2000.
- [13] C. Chappell. IMS's web 2.0 problem. *Light Reading's Services Software Insider*, 2007.
- [14] A. S. E. Burger, J. Van Dyke. Basic network media services with sip. Technical report, ietc - RFC 4240, 2005.
- [15] Eclipse. Graphical modeling framework tutorial, disponible en http://wiki.eclipse.org/index.php/gmf_tutorial. Technical report, 2006.
- [16] D. Ferry and S. Lim. JSR 22. JAIN SLEE API Specification. Technical report, Java Community Process, Mar. 2004.
- [17] R. Guarneri, A. M. Sollund, D. Marston, E. Fossbak, B. Berntsen, G. Nygreen, G. Gylterud, R. Bars, and A. Kerdraon. Report on the state of the art in personalisation, common framework. Technical report, ePerSpace - IST Integrated Project, 2004.
- [18] G. J. Rosenberg, H. Schulzrinne. Sip: Session initiation protocol. Technical report, ietf, 2002.
- [19] F. M. M. Spencer, B. Capouch. Iax: Inter-asterisk exchange version. Technical report, ietf - RFC 5456, 2009.
- [20] S. K. M. Wahl, T. Howes. Lightweight directory access protocol. Technical report, IETF - RFC 2251, 1997.
- [21] A. Moyer, S. Umar. The impact of network convergence on telecommunications software. *Communications Magazine*, 2001.
- [22] P. O'Doherty. Jsr 289 - sip servlet v1.1. Technical report, Java Community Process, 2008.
- [23] E. G. P. Calhoun, J. Loughney. Diameter base protocol. Technical report, IETF - RFC 3588, 2003.

- [24] L. Red HatMiddleware. Mobicents. the open source sllc and sip server. disponible en <http://www.mobicents.org/index.html>, 2009.
- [25] P. C. S. Farrell, J. Vollbrecht. Aaa authorization requirements. Technical report, ietf - RFC 2906, 2000.
- [26] S. Sayer. Sems-ng design overview inside the media server. Technical report, iptego GmbH, 2006.
- [27] B. Sharif. Elastix without tears. Technical report, PaloSanto Solutions, 2008.
- [28] D. Silas. Mobicents media server guide. Technical report, JBoss, 2008.
- [29] VMware Inc. Página web de VMware, disponible en <http://www.vmware.com>, 2009.
- [30] Ángel Cruz. Una nueva convergencia: ¿java en la red? Technical report, 2005.