

Servicios web SOA utilizando agentes inteligentes para la integración de aplicaciones y servicios de las instituciones de educación superior

SOA Web services using intelligent agents for integration of applications and services of institutions of higher education

Mariluz Romero García*

John Javier Angarita Escobar**

Luis Eduardo Reyes Forigua***

Fecha de recepción: 20 de mayo del 2011

Fecha de aceptación: 16 de junio del 2011

Resumen

En este artículo se muestra el proceso y la combinación de la tecnología de agentes inteligentes con la tecnología de servicios web, aplicándola solamente a una tecnología de información para mejorar y tener mayor disponibilidad de contenidos informáticos. En el caso específico del sector universitario, se habla de algunas metodologías significativas de la web semántica, para el uso de agentes inteligentes con servicios web y además se proponen retos para mejorar en el futuro la combinación de estas dos tecnologías (agentes inteligentes y servicios web).

Palabras clave: servicios web, agentes inteligentes, web semántica, metodologías de desarrollo para agentes inteligentes.

* Docente de la Universidad Distrital Francisco José de Caldas, Bogotá D.C, Colombia. Celular: 3115986937. Correo electrónico: mromerog00@yahoo.com

** Estudiante de Ingeniería Telemática, de la Universidad Distrital Francisco José de Caldas, Bogotá D.C, Colombia. Celular: 3003259840. Correo electrónico: john_javes@yahoo.com

*** Estudiante de Ingeniería Telemática, de la Universidad Distrital Francisco José de Caldas, Bogotá D.C, Colombia. Celular: 3003258623. Correo electrónico: avantgarden12@yahoo.com

Abstract

This article shows the process and the combination of intelligent agent technology with web services technology, applied only to information technology to improve and greater availability of computer content. In the case of the university sector, there is talk of some significant methodologies of the Semantic Web for the use of intelligent agents for Web services and also offer challenges to improve in the future the combination of these two technologies (intelligent agents and services site).

Key words: Web services, intelligent agents, semantic web, development methodologies for intelligent agents

Categorías y descripción de sujetos

Servicios Web: "Un Servicio Web es un tipo de aplicación Web". Son aplicaciones modulares autodescriptivas que se pueden publicar, ubicar e invocar desde cualquier punto de la Web o desde el interior de una red local basada en estándares abiertos de Internet (Vasudevan Venu, 2001).

Agentes inteligentes: un agente es un programa de software que puede realizar determinada acción definida para un usuario, con un nivel de inteligencia que le permite desarrollarla de alguna manera autónoma y que puede interactuar con su entorno (Wooldridge, 2002).

Términos generales

Metodología Prometheus: es una metodología para el desarrollo de agentes inteligentes creada por Lin Padgham y Michel Winikoff, en la cual se define un lenguaje de modelado relativamente sencillo, que forma parte de los fundamentos de AUML junto con otras metodologías. Prometheus ha sido desarrollado de manera conjunta con agent oriented software, su ambiente desarrollo PDT (Prometheus Design Tool), actual-

mente está en constante desarrollo (Camargo, s.f., p. 7).

Metodología MCVS (modelo ciclo de vida SOA) con enfoque ágil: con la finalidad de encontrar un enfoque que permita incorporar los principios de arquitectura orientada a servicios en los ambientes de negocio, sin necesidad de esperar que se haya finalizado el proceso en toda la organización, ha surgido el MCVS con enfoque ágil. La modalidad de trabajo de este modelo difiere de otros ampliamente, ya que se ocupa de ejecutar el análisis del negocio en paralelo al diseño de servicios y desarrollo.

Esta forma de trabajo tiene una componente de esfuerzo adicional, con el lógico costo asociado. Esto se debe a la necesidad de tener que ajustar los servicios construidos para alinearlos con los modelos de negocio que pueden ir cambiando a medida que se avanza con el análisis (López, Echeverría y Fierro, 2006).

Agentes inteligentes determinísticos: si el estado siguiente de un ambiente se determina completamente mediante el estado actual, así como las acciones escogidas por los agentes, se dice que el ambiente es determinista. En principio, un agente no tiene por qué preocuparse sobre la incertidumbre en un am-

biente accesible y determinista (Hipola y Vargas-Quesada, 1999).

FIPA: La Foundation for Intelligent Physical Agents (FIPA) es un organismo para el desarrollo y establecimiento de estándares de software para agentes heterogéneos que interactúan y sistemas basados en agentes. FIPA fue fundada como una organización suiza, sin ánimo de lucro, en 1996, con el ambicioso objetivo de definir un conjunto completo de normas para la implementación de sistemas en los que se puedan ejecutar agentes (plataformas de agentes) y especificación de cómo los propios agentes se deben comunicar e interactuar (Blacoe y Portabella, 2005).

Web semántica: el proyecto de la web semántica, impulsada por el W3C (World Wide Web Consortium), trata de crear un medio universal para el intercambio de información, dotándole de semántica a los contenidos de los documentos Web, de forma que estos contenidos sean comprensibles por las máquinas. La web semántica es una visión de una Web con autonomía, en la cual la información no solo es comprensible por los seres humanos, sino que también está disponible de una manera formal para sistemas inteligentes. Si esta visión se fortalece en el futuro, se podrían concebir mejores generaciones de aplicaciones Web, como buscadores semánticos más eficientes o aplicaciones personalizadas o adaptables al usuario y más autónomas (Berners-Lee, Hendler y Lassila, s.f.).

Introducción

El acceso a las tecnologías de la información y su óptima implantación es, sin duda, un aspecto clave para el desarrollo de las organizaciones, dando indudablemente beneficios económicos, sociales y culturales. Actualmente, la web es la mejor plataforma para la implantación de estas tecnologías, pero la

tendencia es que se requerirá cada vez una automatización mayor. Para hablar de esta tendencia, es necesario hablar de web semántica, sus objetivos y las tecnologías que la conforman.

Para lograr un mayor nivel de automatización, las tecnologías que más aportan a este objetivo son los agentes inteligentes, en combinación con los servicios web, debido a que los primeros tienen la capacidad de ser autónomos en la ejecución de procesos, capacidad de aprendizaje y toma de decisiones en el entorno en el que se implementan. Es necesario enfatizar en estas tecnologías, estudiar las metodologías existentes y proponer retos para un constante desarrollo de estas.

Identificación de agentes

Identificación, descripción, funcionalidad de los agentes

En la fase de análisis del aplicativo, se identifican dos agentes inteligentes: un agente para el perfil del usuario de la aplicación, y otro para filtrar las búsquedas del usuario.

Agente perfil usuario

El agente perfil usuario es el encargado de:

- **Recibir información:** este agente recibirá por parte del usuario registrado toda la información para crear un perfil con las tendencias de búsqueda de información dentro de la aplicación; éstas serán guardadas en la ontología.
- **Actualizar perfil usuario:** una vez el usuario registrado realice varias búsquedas dentro del aplicativo, el perfil de usuario será actualizado de acuerdo con estas nuevas búsquedas y será almacenado dentro de la ontología.
- **Enviar información:** el agente perfil usuario enviara la información necesaria

al agente de filtrado para obtener resultados de una búsqueda acorde con el perfil del usuario registrado.

Agente filtro búsquedas

El agente filtro búsquedas es el encargado de:

- Filtrar consulta: el agente cuando tenga conocimiento de cuál es la consulta que desea realizar el usuario registrado y según sus tendencias de búsqueda ya identificadas por el agente perfil usuario, se ejecutará el filtrado de la información recopilada en la búsqueda, organizándola para entregarla al usuario registrado.
- Recibir información: cuando el agente filtro de búsquedas termine su(s) acción(es) y retorne la respuesta correcta, según la consulta realizada por el usuario, el aplicativo recibirá toda la información extraída por dicho agente.
- Mostrar información: al recibir toda la información este agente la suministrará los resultados obtenidos dentro de la interfaz del aplicativo, de una manera organizada y amble con el usuario registrado.

Arquitectura agente inteligente

Agente perfil usuario:

- Tipo: agente reactivo estático.
- Papel: el agente se encarga de capturar información del usuario y crearle un perfil, según las consultas que realiza dentro de la aplicación.
- Posición: dentro de la jerarquía de agentes, este agente tiene la misma ubicación con respecto al agente filtro búsquedas.
- Descripción: el agente de filtrado es de tipo reactivo, cuenta con un sensor que se activa cuando el usuario registrado define algún tipo de búsqueda y actualiza dicho perfil.

Agente filtro búsquedas:

- Tipo: agente reactivo estático.
- Papel: el agente se encarga de filtrar la información obtenida dentro de una búsqueda solicitada por el usuario, según el perfil almacenado por el agente perfil usuario.
- Posición: dentro de la jerarquía de agentes, este agente tiene la misma ubicación con respecto al agente perfil usuario.
- Descripción: el agente de filtrado es de tipo reactivo, cuenta con un sensor que se activa cuando el aplicativo entrega los resultados una búsqueda solicitada por el usuario registrado.

Metodología MCVS (modelo ciclo de vida SOA) con enfoque ágil

Identificación del servicio web

Durante el análisis de requerimientos del aplicativo, se identificó que era necesario diseñar un servicio web para la comunicación de varias bases de datos de las entidades de educación superior a nuestra aplicación, a fin de hacer las consultas sugeridas por el usuario registrado dentro de estas bases de datos y poder mostrar los datos debidamente filtrados dentro de nuestro aplicativo.

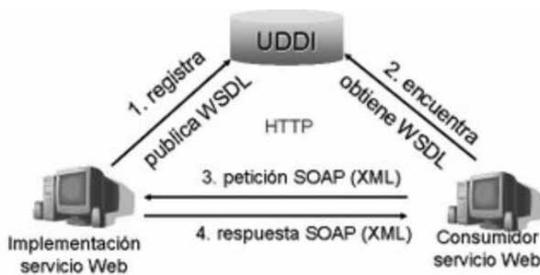
Arquitectura servicio web

La SOA tradicional es aquella que utiliza los principios y tecnologías básicos de los Servicios Web. Esto significa utilizar SOAP como lenguaje de intercambio, WSDL como lenguaje para la descripción de los servicios y UDDI (Universal Description, Discovery and Integration) para la publicación o registro de los mismos [N]. En la figura 1, que se muestra a continuación, se puede ver la estructura básica de funcionamiento de una

SOA tradicional. En la figura, se puede observar la existencia de tres roles claramente diferenciados:

- Cliente del servicio: es el que solicita la ejecución del servicio web y, por tanto, el que lo consume.
- Proveedor del servicio: es el encargado de implementar el servicio web y ofrecerlo a los clientes.
- Registro del servicio: es un repositorio donde se almacenan las descripciones de los servicios, para que así los clientes puedan buscar el servicio web que mejor se adapte a sus necesidades (López, Echeverría, Fierro, 2006).

Figura 1. Estructura básica de funcionamiento de una SOA tradicional



Fuente: <http://www.slideshare.net/Magi20/soa-y-los-sevicios-web-tradicionales>

Diseño e implementación del servicio web

En la etapa de diseño se asignaron las siguientes funciones al servicio web:

- Conectar las diferentes bases de datos de las instituciones de educación superior que se inscribieran en el aplicativo.
- Transmitir y recibir de manera segura los datos que solicita el usuario registrado.
- Facilitar la integración de varias tecnologías por medio del servicio web.

Para la implementación del servicio web, se monta un servidor de aplicaciones Jboss, el cual soporta el servicio web y los agentes inteligentes. El servicio web está escrito en lenguaje Java, puesto que por configuración del servidor es el lenguaje con que tiene más compatibilidad.

Desarrollo de la aplicación

En el desarrollo del aplicativo final, intervienen varios componentes que lo conforman, como la base de datos, el servidor de aplicaciones, la plataforma en la que se montan todos los servicios y la interfaz gráfica. Para cada uno de ellos, se debe tener en cuenta la aplicación en la cual va a ser desarrollada e implementada, que estas plataformas individuales para cada fin, sean compatibles entre sí y que puedan trabajar como uno solo, para que al final, la experiencia del usuario sea de lo más agradable y productiva. Para lograr tal objetivo, se utilizaron las herramientas más convenientes, que a continuación enumeramos.

Componentes de la aplicación

Base de datos

Para el desarrollo de la base de datos, se seleccionó MySQL, debido a su flexibilidad y compatibilidad con las demás herramientas de desarrollo, además de ser una plataforma libre. MySQL es un sistema de gestión de bases de datos relacional, multihilo y multiusuario, actualmente lo desarrolla Sun Microsystems, bajo la licencia de software libre. La base de datos diseñada para este proyecto es relacional, en la cual se compila toda la información personal de los usuarios registrados, usuarios universidad, conexiones de los servicios web, historial de búsquedas y archivos disponibles para descargar.

Eclipse

El entorno de desarrollo utilizado para la implementación del aplicativo fue EasyEclipse en la versión 1.2.2.2. La implementación se llevó a cabo desde la perspectiva J2EE. La instalación del entorno de desarrollo se realiza con un asistente y una vez se ha instalado se puede iniciar desde un acceso directo o desde el directorio en el que se halla instalado. En este aplicativo se integró la plataforma MySQL y el servidor de aplicaciones Jboss, a la vez que es el soporte de los agentes inteligentes y la interfaz gráfica del aplicativo. Eclipse es la plataforma en la cual va soportado todo el proyecto.

JBoss

El servidor de aplicaciones utilizado para el despliegue de la aplicación es JBoss en la versión 4.0.2, se utilizó este servidor puesto que es compatible con la mayoría de herramientas para agentes y es una herramienta de software libre que gestiona de forma óptima las aplicaciones, además es un servidor de aplicaciones J2EE de código abierto implementado en Java puro e implementa todo el paquete de servicios que ofrece J2EE. Al estar basado en Java, JBoss puede ser utilizado en cualquier sistema operativo que lo soporte. En el aplicativo JBoss, está soportado el servicio web y las conexiones a las bases de datos, tanto a la local como a las de las instituciones educativas que se registren en el aplicativo.

Interfaz gráfica

Para el diseño de la interfaz gráfica de la aplicación, se instaló el software editor Adobe Dreamweaver CS4, debido a su gran facilidad para administrar código php y su integración con bases de datos en MySQL, además de la posibilidad de trabajar con base en

plantillas preestablecidas, diseños atractivos y de fácil uso para el usuario. La codificación de esta interfaz se realizó en el lenguaje php, ya que ofrece gran flexibilidad y compatibilidad con los distintos navegadores actuales de Internet, creando una interfaz muy liviana, funcional y amable con el usuario.

Resultados parciales

La aplicación hasta la fecha aún está en etapa de desarrollo, toda la plataforma Eclipse, Jboss, MySQL y Adobe Dreamweaver CS4 está instalada y configurada como servidor web, y desde allí se puede acceder a la aplicación por medio de la siguiente dirección:

<http://201.244.221.128:8080/webserviceu/index.htm>

El servicio web está debidamente configurado y se conecta a otras bases de datos que creamos para pruebas, la interfaz gráfica ya está completa y se puede navegar por todo el aplicativo, crear nuevos usuarios, se pueden agregar universidades y crear la conexión entre el aplicativo y su plataforma. Se tuvo en cuenta un esquema de colores que relaje la vista del usuario, que mantenga la atención sobre la información que este está buscando, que sea limpia y fácil de manejar, a la vez que se seleccionaron una fuente de letras claras, generando un buen contraste con el fondo de la interfaz. Los agentes inteligentes están en una etapa temprana de desarrollo, ya están identificados y tienen sus tareas y comportamiento definidos, además de su arquitectura y roles dentro de la aplicación.

Al aplicativo en conjunto se le han hecho varias pruebas de navegación por parte de varios usuarios, en los que se han tomado en cuenta sus comentarios y las opiniones, con el objetivo de mejorar la aplicación y refinar la interfaz gráfica, como siempre buscando la mejor experiencia para el usuario.

Durante el desarrollo de los agentes inteligentes, se deberá refinar las búsquedas de los usuarios registrados, crear sus perfiles, así como deberá guardarlos y actualizarlos dentro de la ontología que se está diseñando, y de esta manera ofrecer el mejor producto terminado, esperando a que sea la base de muchos más en un futuro.

Agradecimientos

Agradecemos por todo su compromiso, colaboración y asesoría a nuestra tutora del proyecto, Ing. Mariluz Romero García, igualmente a todos por su dedicación al Ing. Nelson Becerra, Ing. Roberto Salas y al Ing. Jorge Enrique Rodríguez, así como a nuestras familias y a la Universidad Distrital Francisco José de Caldas por esta oportunidad.

Referencias

- Berners-Lee, T., Hendler, J. y Lassila, O. (s.f.). *The Semantic Web*. 2001, Scientific American. Recuperado de: DOI= http://kill.dev.c.at/system/files/scientific-american_0.pdf
- Blacoe, I. y Portabella, D. (2005). *Guidelines for the integration of agent-based services and web-based services*.
- Camargo, M.D. (2007) Capítulo 3 Tecnología de agentes inteligentes y la metodología Prometheus. En *Sistema de selección de personal inspirado en agentes inteligentes* Recuperado de: http://catarina.udlap.mx/u_dl_a/tales/documentos/lis/camargo_m_d/capitulo3.pdf.
- Hipola, P. y Vargas-Quesada, B. (1999). *Agentes inteligentes: definición y tipologías. Los agentes de información*. El profesional de la información, 1999.
- López, G., Echeverría, A., Fierro, P. (PhD.), Jeder, I. (2006). *Una propuesta de modelos de ciclo de vida (mcs) para la integración de los procesos de negocio utilizando Service Oriented Architecture (SOA)*. Recuperado de: <http://www.ing.unp.edu.ar/wicc2007/trabajos/ISBD/116.pdf>
- Vasudevan, V. (2001, abr.). *A Web Services Primer*. Recuperado de: <http://www.xml.com/pub/a/2001/04/04/web-services/index.html>
- Wooldrige, M. (2002). *An Introduction of Multi Agent Systems*. John Wiley & Sons.