

# Enfoque para la aplicación de Semantic Web Services en Sistemas Centrados en el Usuario

Fernando Daniel Sanabria<sup>\*1</sup>, M. Ing. Jorge Salvador Ierache<sup>\*1</sup>

Instituto de Sistemas Inteligentes y Enseñanza Experimental de la Robótica  
Facultad de Informática, Ciencias de la Comunicación y Técnicas Especiales, Universidad de Morón  
Cabildo 134 Morón (1708), Provincia de Buenos Aires, Argentina Tel: 54-11-5627-2000 \*1  
e-mail: fersanabria@yahoo.com.ar - jierache@yahoo.com.ar

## RESUMEN

Los servicios basados en la Web ya se han convertido en una herramienta más en la vida de la mayoría de las personas. En la actualidad por ejemplo se utiliza la Web para pagar servicios, comprar artículos o escuchar programas de radio. Todo indica que esto seguirá expandiéndose y que cada vez habrá más aplicaciones para que los usuarios puedan acceder a más servicios. La utilización Semantic Web Services (SWS) básicamente permite que diferentes servicios desplegados en la Web puedan interactuar de manera automática a un nivel semántico, basándose en las descripciones de interfaces usando ontologías.

Este trabajo analiza los aspectos más relevantes para lograr un enfoque de SWS aplicado a Sistemas Centrados en el Usuario (C-U), y se propone una solución para uno de los elementos fundamentales de este enfoque, como es un Servidor de Modelos de Usuario (SMU).

**Palabras clave:** Semantic Web Service, usuario, aplicación, ontología, modelo de usuario.

## 1. INTRODUCCION

Es un objetivo de la comunidad informática en general que las aplicaciones puedan conformarse de manera dinámica para responder de la forma más precisa posible a los requerimientos de un usuario en un momento dado. Este es un objetivo, si bien realizable, un tanto ambicioso para poder ser alcanzado en el corto plazo.

Basándose en anotaciones semánticas tomadas de Semantic Web [1], Semantic Web Services [2] es un enfoque de middleware que apunta a posibilitar que diferentes agentes o aplicaciones convencionales puedan conformar soluciones de manera dinámica y circunstancial para responder a un requerimiento específico dado. Esto se lograría mediante ciertas facilidades como descubrimiento, interacción ó composición de Web Services de manera dinámica [8]. Por otro lado, desde hace ya más de dos décadas [7], se viene trabajando sobre diferentes formas de modelar las características del usuario para que puedan ser bien conocidas por las aplicaciones con las que éste interactúa. Sumar el potencial de estos dos aspectos en los que se viene trabajando en el último tiempo (SWS y Modelos de Usuario), y hacerlos parte de un mismo enfoque, es el objetivo propuesto en este

trabajo. Esto implicaría dos ventajas importantes: poder realizar composición de servicios de manera totalmente dinámica y evitar al usuario realizar tareas operativas, rutinarias o reiterativas, ya que las aplicaciones responderían a sus requerimientos tomando en cuenta sus preferencias de manera anticipada y convirtiéndolo en algo transparente para el mismo. Además, basándose en el conocimiento que incrementalmente irán ganando sobre el usuario, en diferentes situaciones, podrán hacer propuestas de manera activa a éste asistiéndolo en su interacción con la aplicación y en sus decisiones.

En la próxima sección se presenta una introducción y relevamiento del estado del arte sobre SWS y Modelos de usuario, por ser estos los pilares del enfoque desarrollado en este trabajo. En la sección 3 se presenta un contexto del problema identificado con todos los aspectos involucrados. Este contexto es necesario para proveer una idea clara del enfoque tomado y luego en la sección 4 se desarrolla la solución propuesta.

## 2. SEMANTIC WEB SERVICES Y MODELOS DE USUARIO

### SWS

Siguiendo a A. Svirskas, M. Wilson [4], mientras que el advenimiento de las arquitecturas SOA basadas en Web Services han presentado una buena base con respecto a middleware basado en la Web, todavía hay algunos aspectos que no han sido cubiertos y que quizás no puedan ser solucionados por los “ingredientes” clásicos de Web Services, tales como SOAP, Web Service Description Language (WSDL), Universal Description, Discovery and Integration (UDDI). Crear aplicaciones de software requiere tiempo y esfuerzo de especialistas de IT calificados. Aún mientras prácticas de ingeniería de software bien probadas son aplicadas de forma extensiva, existe todavía un cierto vacío entre las intenciones y deseos de los usuarios y las funcionalidades ofrecidas por las aplicaciones de software desarrolladas [3]. Es un deseo aún la posibilidad de que, por ejemplo, agentes puedan, acorde con requerimientos del usuario o el negocio, configurar aplicaciones “on-the-fly” basándose en componentes y servicios disponibles en la Web. Para hacer uso de un Web Service, un agente necesita una descripción del servicio interpretable por un programa de computadora y todos los instrumentos necesarios para poder hacer un “matching” dinámico

del Web Service. Si queremos avanzar hacia descripciones más enriquecidas de los Web Services, la Web Semántica es la opción evidente. Como describió T. Berners-Lee [1] “La Web semántica es una extensión de la Web actual en la cual se le da un significado bien definido a la información, posibilitando un trabajo cooperativo mejorado entre personas y computadoras”. Un Semantic Web Service describe sus propiedades y capacidades a fin de que aplicaciones de software puedan determinar automáticamente su propósito. Las Ontologías para servicios Web deberían hacerlos entendibles por máquinas y soportar ciertas características fundamentales como descubrimiento de servicios automático, invocación, composición. En resumen, permitir una interoperabilidad automática para que un conjunto de servicios puedan cumplir un determinado objetivo. De esta manera, entidades de software pueden crear soluciones completamente nuevas, las cuales son parte de colaboraciones de negocio on-demand.

En la Figura 1 se pueden apreciar las diferentes dimensiones en las que se pueden estructurar los componentes de Semantic Web Services. Estas dimensiones son: Actividades, Arquitectura y Ontología de servicio[8].

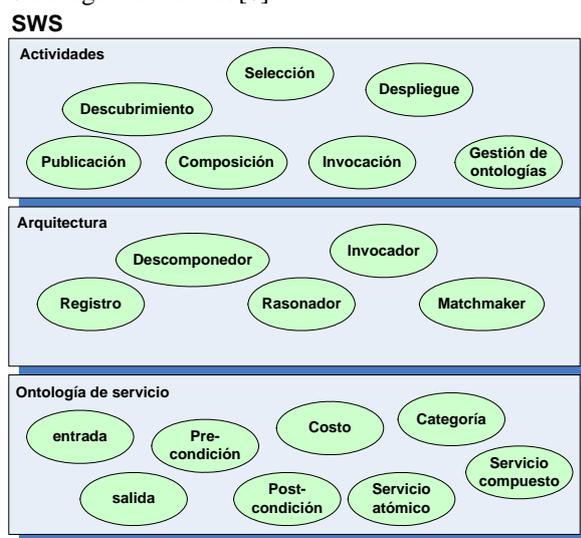


Fig. 1 Dimensiones de SWS

### Modelado de Usuario

El modelado de usuario tiene sus comienzos por fines de la década del 70' cuando Allen, Cohen y Perrault presentan los primeros trabajos basados en esta idea (p. E., Perrault, 1978; Cohen y Perrault, 1979; Allen, 1979). En estos comienzos en el modelado de usuario, el modelado se realizaba en la aplicación misma y no existía una clara distinción entre los componentes que administraban la información del usuario y los componentes que realizaban otras tareas.

A partir de mediados de los años ochenta, tal separación se hizo cada vez más visible (p. e., Kobsa, 1985; Sleeman, 1985; Kass, 1988; Allgayer, 1989), pero no se puso énfasis en la modularización de los

componentes de modelado de usuario para permitir que sean reutilizables en nuevos sistemas “user-adaptive”.

Ya para fines de los 90' el valor de la personalización en la Web es altamente reconocido en el área del comercio electrónico (Hof et al., 1998; Allen et al., 1998; Cooperstein et al., 1999; Hagen et al., 1999). La característica central de los sistemas de modelado de usuario a partir de ese momento fue la modalidad servidor, en la cual estos no se encuentran integrados en las aplicaciones mismas sino que las sirven a través de comunicación entre procesos [9, 10]. De esta manera pueden proveer información a más de una aplicación de manera centralizada.

La personalización, interés fundamental de la comunidad de modelado de usuario, refiere a métodos para ganar algún conocimiento acerca de los usuarios, representados como un modelo de usuario, y explotando ese modelo para ajustar el comportamiento de los sistemas a las necesidades de los individuos. Gracias al aporte de la Web Semántica los modelos de usuario pueden disponer hoy de estándares bien definidos y ontologías, para proveer extensibilidad, flexibilidad, interoperabilidad y reusabilidad.

### 3. EL PROBLEMA ENCONTRADO

En esta sección se intenta presentar de manera clara el contexto, con todos sus aspectos, que da origen al enfoque desarrollado en el trabajo. Entre estos aspectos está el de un middleware semántico y un servidor de modelos de usuario. Para introducirnos de manera clara, vamos a describir algunas situaciones factibles de la vida real de un usuario como ejemplo.

Consideremos un usuario planificando y ejecutando un viaje de placer por Miami para sus vacaciones. Lo primero que nuestro usuario deseará hacer son todas las reservas de vuelos, hospedaje y automóvil, para lo cual necesitaría realizar las búsquedas, reservas e ingresos de datos reiterados necesarios con todo el trabajo reiterativo y tedioso que esto implica. Pensemos ahora en una aplicación de planificación de viajes centrada en el usuario en la cual el usuario ingresa origen y destino de vuelo, fechas, costos, cómo espera que esté el tiempo en su estadía, comodidades del hotel en el cual le gustaría hospedarse y qué tipo de auto desea rentar para moverse por aquella ciudad. Todo esto desde un solo punto de entrada y nuestra aplicación se encargará de hacer la búsqueda con estos requisitos y presentarle al usuario diferentes opciones para que él pueda tomar la decisión final. Una vez que se decide sólo debe aceptar y se toman automáticamente sus datos personales, número de tarjeta de crédito y otros datos requeridos, de su modelo de usuario digital personal.. Luego pude recibir por correo los boletos y otra documentación necesaria o puede retirarla personalmente según haya elegido. Una vez en Miami ya lo estará esperando su auto rentado y su habitación reservada. Se podría seguir describiendo como el

usuario, acompañado de su PDA, sigue utilizando servicios personalizados gracias a un servicio de Internet wireless y aplicaciones centradas en su perfil y preferencias. A continuación se identifican y describen brevemente los aspectos más importantes que se consideran relevantes para un enfoque de SWS en Sistemas Centrados en el Usuario.

#### **Middleware de Semantic Web Services**

En la actualidad podemos estar hablando de todas las bondades que tienen los Semantic Web Services y lo vemos como una realidad no tan lejana gracias a que muchas instituciones, universidades, empresas, comunidades, están dedicando un gran esfuerzo en investigación y desarrollo. El avance demostrado en este enfoque en el último tiempo hará que en un futuro cercano podamos basarnos en una interoperabilidad estandarizada, sencilla y robusta a niveles semánticos a la hora de diseñar soluciones futuras [2, 6].

#### **Un modelo de usuario**

Para que diferentes aplicaciones puedan responder tomando en cuenta el perfil y las preferencias de un determinado usuario, debe haber un lugar común en donde estén almacenados estos datos del usuario y puedan ser accedidos por las aplicaciones que los requieran. Representar las características, atributos, preferencias de un usuario de manera abstracta implica hacer un modelo de éste. Un modelo de usuario mantenido en un servidor al que puedan acceder diferentes “clientes” para consultas y actualizaciones constantes es un elemento fundamental para poder centrar en el usuario las aplicaciones distribuidas en la Web. El modelo del usuario podría actualizarse y consultarse por diferentes mecanismos, y de esta manera crece el conocimiento que se tiene del mismo.

#### **Semantic Web Services desplegados**

La cantidad y diversidad de servicios publicados crea un entorno propicio para un mercado electrónico extensivo en Internet. En este entorno tienen lugar aplicaciones Centradas en el Usuario.

#### **Aplicaciones para negocios o problemas específicos**

Para hacer un uso de los Web Services desplegados en la red en un negocio específico o para resolver un determinado problema, necesitamos una entidad que “orqueste” estos servicios de la manera adecuada para llegar a un objetivo determinado. Para responder en forma personalizada a los objetivos del usuario, debe haber aplicaciones que sean el punto de entrada para los requerimientos de estos y que se encarguen de hacer las composiciones de Web Services que sean necesarias. Estas aplicaciones que llamamos Centradas en el Usuario (C-U) son la “fachada” que el usuario ve y por lo tanto con la única entidad con la que interactúan cuando persiguen un objetivo determinado o un conjunto de objetivos fuertemente relacionados. Puede haber aplicaciones de mayor capacidad de decisión y autonomía implementadas con agentes, otras que tengan procesos corriendo en batch para procesar información que luego

presentarán al usuario o también puede haber aplicaciones del estilo Web clásico.

Además de estos aspectos fundamentales, existen otros aspectos complementarios que deben tenerse presentes para abarcar la problemática en su totalidad. Estos son el método de cobro en un entorno e-business y las interfaces de usuario adaptables a las características del hardware.

Los Web Services desplegados en Internet, se encuentran allí porque obtienen algún beneficio económico por el servicio que proveen. En los casos en que por el servicio provisto la empresa no obtiene ningún tipo de rédito económico, el beneficio puede obtenerse por el cobro del uso de un Web Service. Un ejemplo de esto es un Web Service que provea información meteorológica. Para poder gestionar el cobro de consumo de Web Services será necesario un mecanismo y estándares bien claros que definan cómo se realizarían esas operaciones en un entorno e-business.

Para lograr que las aplicaciones sean accesibles de igual manera desde cualquier dispositivo y en toda su funcionalidad se debe tomar en cuenta que las características de accesibilidad de cada dispositivo varían, y en algunos casos ampliamente. Estas diferencias en la interacción con el usuario no deben tener impacto en la lógica de negocio de las aplicaciones sino que deben manejarse de manera transparente tanto para la aplicación como para el dispositivo del usuario.

## **4. SOLUCION PROPUESTA**

Luego de identificar todos los aspectos a tomar en cuenta en el enfoque propuesto, a continuación en la Figura 2 se presenta un diagrama conceptual de la solución propuesta. Aquí se puede apreciar básicamente cómo interactúan los diferentes elementos participantes.

De todos los elementos que se identifican en el diagrama, este trabajo se focaliza en el Servidor de Modelos de Usuario (SMU). Entendemos que el SMU básicamente debe cumplir con los siguientes requisitos:

- Ser totalmente independiente y auto contenido
- La información del usuario es mantenida de manera centralizada o virtualmente integrada en un repositorio y puesta a disposición de más de una aplicación al mismo tiempo.
- La información de usuario adquirida por una aplicación puede ser usada por otra y viceversa
- La información del usuario es almacenada de manera no redundante. La consistencia y coherencia de la información recopilada por las diferentes aplicaciones pueden ser alcanzadas más fácilmente.
- Posibilitar la formación de grupos y estereotipos.
- Presentar una interfase de acceso bien clara y definida
- Permitir su migración con total independencia del hardware

- Ser multi-plataforma
- Proveer métodos de backup y restore de toda la información mantenida
- Aplicar las normas y configuraciones de seguridad de la información de manera centralizada sobre los servidores de modelos de usuario.
- Contar con una documentación clara y completa.

Los pilares de la arquitectura del SMU, para poder cumplir con estas características, son principalmente: el lenguaje de ontologías OWL, utilizado para mantener el modelo del usuario almacenado y aplicar las reglas de mapeo correspondientes, y las interfaces basadas en Web Services para permitir un acceso estandarizado a los clientes del SMU.

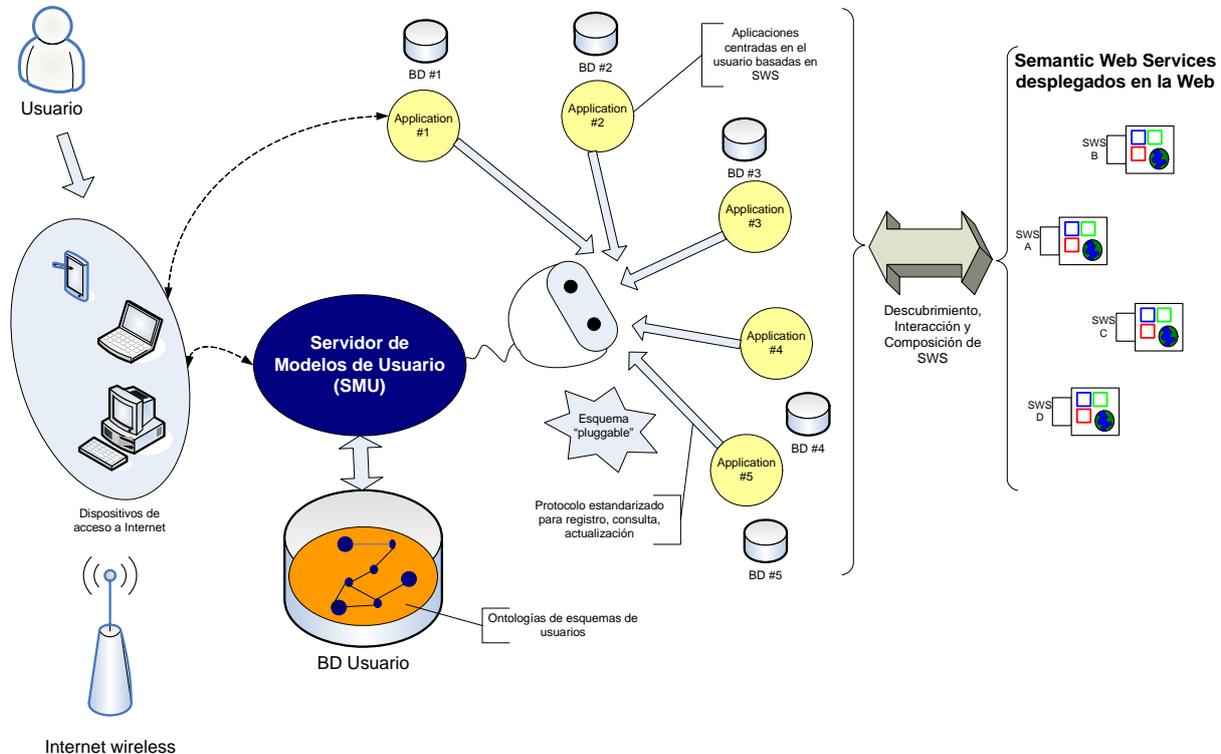


Figura 2. Diagrama general de la solución propuesta

### Ontologías de usuarios

El lenguaje OWL[11] nos permite describir las características del usuario, en un modelo de éste, utilizando descripciones semánticas para sus atributos. Cada aplicación que utilice el SMU tendrá su propio esquema de usuario, el cual crea, mantiene y consulta. Luego a un nivel más elevado de abstracción existen esquemas de modelos de usuario que permiten realizar mapeos entre los esquemas de cada aplicación en particular. Todos los esquemas mantenidos por el SMU, para las diferentes aplicaciones C-U que lo utilizan, son mapeados a un modelo más abstracto, general y depurado que incrementa su información de forma exponencial con el uso de las diferentes aplicaciones C-U. Esto permite que cuando una nueva aplicación se suscribe al SMU y crea su esquema, éste ya pueda ser cargado inicialmente con cierta información del usuario que se obtiene desde el modelo general que el SMU ya tiene del mismo.

### Interfaces de acceso basadas en Web Services

Las interfaces basadas en Web Services son altamente propicias para permitir que el SMU se inserte como un componente independiente y auto contenido en entornos abiertos y cambiantes. Los clientes ven al SMU como un proveedor de servicios relacionados

con información del usuario, al cuál acceden por interfaces bien definidas. Las especificaciones utilizadas para la implementación de las interfaces del SMU son WSDL[12] y SOAP[13]. La implementación del SMU y sus interfaces deben estar basadas en tecnologías multi-plataforma, para lograr una total independencia del hardware y software de base en el que eventualmente pueda estar corriendo.

## 5. CONCLUSIONES

Se identificó la necesidad de ir hacia un enfoque que integre diferentes temas de estudio, que actualmente se están desarrollando en la comunidad informática, con el fin de lograr aplicaciones aptas para entornos abiertos y que se centren en las características del usuario. Entre estos temas en desarrollo se destacan como fundamentales los SWS y los Modelos de usuario[14, 15]. Luego de identificarse el contexto del problema con todos sus elementos e interrelaciones, se debe continuar hacia el diseño de soluciones para cada uno de estos aspectos, concluyendo con una integración y el diseño detallado completo del enfoque. Este trabajo es el origen para seguir profundizando en el diseño de la arquitectura para un Servidor de Modelos de Usuario.

## 6. REFERENCIAS

- [1] T. Berners-Lee, et al. The Semantic Web, Scientific American, May 2001.
- [2] Semantic Web Service Initiative (SWSI). [www.swsi.org](http://www.swsi.org)
- [3] J. Domingue, L. Cabral, et al. "A Position statement on IRS-III: A Comprehensive Approach to Creating and Using Semantic Web Services", W3C workshop on Frameworks for Semantics in Web Services, 2005.
- [4] A. Svirskas, M. Wilson, et al. "Towards an Efficient, Reliable and Collaborative Web: from Distributed Computing to Semantic Description, Composition and Matchmaking of Services", W3C workshop on Frameworks for Semantics in Web Services, 2005.
- [5] M. Wagner, M. Paolucci. "Enabling Personal Mobile Applications through Semantic Web Services", 2005.
- [6] Data, Information, and Process Integration with Semantic Web Services (DIP). [dip.semanticweb.org/index.html](http://dip.semanticweb.org/index.html)
- [7] A. Kobsa, Generic User Modeling Systems, 2000.
- [8] L. Cabral, J. Domingue, et al. Approaches to Semantic Web Services: An Overview and Comparisons, 2004.
- [9] K. van der Sluijs, G. Houben . "Towards a Generic User Model Component", Workshop on Personalisation on the Semantic Web PerSWeb05, 2005.
- [10] D. Heckmann, T. Schwartz, et al. GUMO – the General User Model Ontology, 2005.
- [11] OWL Web Ontology Language Overview W3C Recommendation 10 February 2004. [www.w3.org/TR/owl-features/](http://www.w3.org/TR/owl-features/)
- [12] Web Services Description Language (WSDL) 1.1 W3C Note 15 March 2001. [www.w3.org/TR/wsdl](http://www.w3.org/TR/wsdl)
- [13] SOAP Version 1.2 W3C Recommendation 24 June 2003. [www.w3.org/TR/soap/](http://www.w3.org/TR/soap/)
- [14] User Modeling Inc. (UM Inc.) Home Page. [um.org](http://um.org)
- [15] Ubiquitous User Modeling. [www.u2m.org](http://www.u2m.org)