

# UPSranker: Integrando Programación con Restricciones y Evaluación de Reglas para el Ranking de Servicios Web Semánticos

José María García, David Ruiz, Pablo Fernandez, y Octavio Martín-Díaz\*

Universidad de Sevilla

E.T.S. Ing. Informática, Av. Reina Mercedes s/n, 41012 Sevilla, España  
{josemgarcia, druiz, pablofm, omartindiaz}@us.es

**Resumen** El descubrimiento y ranking de servicios Web semánticos suele realizarse en base a una serie de preferencias de usuario, descritas usando un formalismo concreto que condiciona tanto la expresividad de las preferencias como la implementación subyacente. Por tanto, para permitir el uso de descripciones avanzadas de preferencias de usuario, se hace necesaria la integración de distintas herramientas basadas en diferentes formalismos. En este trabajo presentamos UPSranker, una herramienta que integra en un componente de ranking para servicios descritos en WSMO la evaluación de reglas y el uso de la programación con restricciones para habilitar la descripción de preferencias mediante funciones de utilidad. El objetivo primordial de esta herramienta es servir como prueba de concepto para la integración de diferentes formalismos en el ranking de servicios, lo que posibilita la extensión de las descripciones de preferencias para mejorar su capacidad expresiva.

## 1. Introducción

Las soluciones actuales para el ranking de servicios web semánticos suelen estar fuertemente acopladas al formalismo escogido para la descripción de las preferencias de usuario, las cuales determinan cuál es el mejor servicio para dicho usuario de entre los previamente descubiertos en un repositorio. Debido a esto, la expresividad de las preferencias queda fijada por dicho formalismo, imposibilitando el uso de técnicas más expresivas, como puedan ser las funciones de utilidad.

De cara a explorar el uso de distintos formalismos para la descripción de preferencias de usuario, UPSranker integra la evaluación de reglas de programación lógica con la resolución de problemas de satisfacción de restricciones. De esta forma, nuestro componente extiende la ontología Web Service Modeling Ontology (WSMO) [6] añadiendo la posibilidad de expresar preferencias de usuario en base a funciones de utilidad [4].

UPSranker está disponible, junto con información detallada y documentación, en <http://www.isa.us.es/upsranker>

---

\* Este trabajo ha sido parcialmente subvencionado por la Comisión Europea y el Gobierno Español bajo los proyectos CICYT Web-Factories (TIN2006-00472) y SETI (TIN2009-07366); y por la Junta de Andalucía bajo el proyecto ISABEL (TIC-2533).

## 2. Características principales de la herramienta

La implementación de la herramienta de ranking realizada, aún una serie de características principales, que permiten evaluar las ventajas de la solución propuesta. Estas características son las siguientes:

- **Extensión de WSMO.** El componente desarrollado se basa en una extensión de la ontología WSMO presentada en [4], donde se proporcionan medios para expresar preferencias mediante funciones de utilidad.
- **Integración de distintos formalismos.** La integración realizada en el proceso de desarrollo de UPSranker se ha materializado, por un lado, en el uso del razonador IRIS<sup>1</sup> para evaluar las reglas de programación lógica de las preferencias utilizadas; y por el otro lado, en la traducción a un problema de satisfacción de restricciones de dichas preferencias [7], cuyo resultado se calcula mediante la librería de programación con restricciones Choco<sup>2</sup>. Sin embargo, la implementación desarrollada permite el uso de otros razonadores de manera transparente, permitiendo desplegar una arquitectura híbrida para el descubrimiento y el ranking de servicios [3].
- **Implementación sobre componentes de WSMX.** UPSranker ha sido desarrollado en base a un componente existente dentro de la implementación de referencia de WSMO, WSMX<sup>3</sup>. Esto permite su integración dentro de este framework de ejecución de servicios web semánticos, pudiendo por tanto utilizarse en entornos reales.

## 3. Extensiones y Trabajo Futuro

Hemos identificado distintas extensiones al componente desarrollado de cara a un posible trabajo futuro. En primer lugar, la extensión de WSMO ya introducida en [4] y utilizada para nuestro componente, está siendo actualmente ampliada y generalizada para poder expresar preferencias de usuario de una manera mucho más natural e intuitiva, teniendo en cuenta la usabilidad del modelo de preferencias desarrollado. Este modelo de preferencias será posible integrarlo en soluciones ligeras de descubrimiento y ranking de servicios, como la desarrollada en el proyecto SOA4All [1]. Además, dicho modelo es compatible con el uso de lenguajes de consulta como SPARQL [5] para obtener búsquedas más eficientes [2].

Por otro lado, respecto a la implementación desarrollada, se pretende continuar el desarrollo de UPSranker para permitir el uso de más formalismos o librerías externas según los elementos descriptivos disponibles para definir preferencias de usuario en un escenario de uso concreto. Asimismo, mediante la generalización del modelo de preferencias antes citada, se desarrollará una evolución hacia un componente adaptable que pueda ser incorporado en buses de servicio o ESBs específicos, como por ejemplo PEtALS<sup>4</sup> o FAST<sup>5</sup>, de manera que pueda proveer servicios de infraestructura a dichos

<sup>1</sup> <http://sourceforge.net/projects/iris-reasoner/>

<sup>2</sup> <http://www.emn.fr/x-info/choco-solver/>

<sup>3</sup> <http://www.wsmx.org>

<sup>4</sup> <http://petals.ow2.org/>

<sup>5</sup> <http://www.isa.us.es/fast>

buses para permitir el descubrimiento y ranking de servicios web semánticos en base a un modelo completo de preferencias de usuario.

#### 4. Conclusiones

En este trabajo se ha presentado UPSranker, un componente capaz de realizar un ranking de servicios web semánticos en base a preferencias de usuario definidas mediante funciones de utilidad. La solución expuesta hace uso de diferentes formalismos, como son la evaluación de reglas y la programación con restricciones, presentando una solución de integración que aporta medios para aumentar la expresividad de las preferencias de usuario utilizadas para el descubrimiento y ranking de servicios web semánticos descritos en WSMO. Asimismo se han analizado las posibles extensiones de este trabajo, exponiendo los diferentes esfuerzos que se han planificado al respecto.

**Agradecimientos.** Los autores desean agradecer a Ioan Toma, de la Universidad de Innsbruck, su apoyo y ayuda para la integración e implementación del componente presentado en este trabajo.

#### Referencias

1. John Domingue, Dieter Fensel, and Rafael González-Cabero. Soa4all, enabling the soa revolution on a world wide scale. In *ICSC '08: Proceedings of the 2008 IEEE International Conference on Semantic Computing*, pages 530–537, Washington, DC, USA, 2008. IEEE Computer Society.
2. J. M. García, C. Rivero, D. Ruiz, and A. Ruiz-Cortés. On Using Semantic Web Query Languages for Semantic Web Services Provisioning. In *The 2009 International Conference on Semantic Web and Web Services (SWWS)*. CSREA Press, 2009.
3. J. M. García, D. Ruiz, A. Ruiz-Cortés, O. Martín-Díaz, and M. Resinas. An hybrid, QoS-aware discovery of semantic web services using constraint programming. In B. Krämer, K.-J. Lin, and P. Narasimhan, editors, *ICSOC 2007*, volume 4749 of *LNCS*, pages 69–80. Springer, 2007.
4. J. M. García, I. Toma, D. Ruiz, A. Ruiz-Cortés, Y. Ding, and J. M. Gómez. Ranking semantic web services using rules evaluation and constraint programming. In *JSWEB 2008*, 2008.
5. Eric Prud'hommeaux and Andy Seaborne. SPARQL Query Language for RDF. Technical report, W3C, 2006.
6. D. Roman, H. Lausen, and U. Keller (Ed.). Web service modeling ontology (WSMO). Working Draft D2v1.4, WSMO, 2007. Available from <http://www.wsmo.org/TR/d2/v1.4/>.
7. A. Ruiz-Cortés, O. Martín-Díaz, A. Durán-Toro, and M. Toro. Improving the automatic procurement of web services using constraint programming. *Int. J. Cooperative Inf. Syst.*, 14(4):439–468, 2005.