

CC71K: Técnicas de Bases de Datos para la Web

Semantic Matching of Web Services Capabilities

Javier González Nicolini

Martes 12 de Noviembre de 2002

1. Abstract

Los autores del paper presentan una visión de la web evolucionando de colección de páginas a colección de servicios interoperantes en la web. Esta interoperación se logrará mediante un match semántico entre descripciones declarativas del servicio requerido y el servicio ofrecido. Actualmente este match semántico está mas allá de las capacidades de los buscadores/directorios de servicios actuales (WSDL/UDDI). Por lo tanto, los autores proponen una solución basada en DAML-S.

2. Motivación/Introducción

La evolución de la Web a una colección de servicios interoperantes implica intercambio de información "a pedido". Esta capacidad de operar con información a pedido le da una gran oportunidad de potenciarse a los e-business, y en particular manejar automáticamente su cadena de suministros. Este manejo automático genera 2 desafíos: Los servicios web deben poder encontrar e interactuar con otros servicios, y además deben interoperar para lograr servicios compuestos. El paper aborda el primer problema propuesto.

Los estándares actuales no son aptos para resolver este problema. SOAP y WDSL, por ejemplo, se preocupan mas de describir mecanismos de transporte e interfaces; UDDI describe entidades por atributos y servicios que provee. Sin embargo, esta descripción es para humanos. Ninguna de las herramientas actuales permita buscar automáticamente servicios en base a lo que proveen.

Esta limitación nace de la falta de una semántica explícita, dado que una búsqueda de capacidades solo podran ser exitosa a un nivel semántico. Es necesario este nivel semántico porque el requeridor de un servicio no sabe a priori que servicios están disponibles, y además un proveedor y un requeridor de un servicio pueden tener distinta perspectiva sobre éste.

Sin embargo, tomando este match semántico *es poco realista pensar en un match exacto*. Por lo tanto, es necesario un *Matching Engine* que pueda, mediante la comprensión semántica del pedido y de la oferta, reconocer el grado de mismatch y quedarse con el servicio más adecuado a las necesidades.

Para esto se usa DAML-S, estrechamente relacionado con DAML+OIL. Soporta la necesidad de representación semántica, razonamiento de subsumas en taxonomías y relaciones. De este modo permite la descripción de una amplia gama de servicios y permite la comparación de éstos.

3. Perfiles DAML-S

Con DAML-S se pueden crear Perfiles de Servicio, que describen la funcionalidad de un servicio web. Se representa con una ontología de 3 divisiones lógicas:

1. ACTOR: Información acerca del proveedor del servicio.
2. ATRIBUTOS FUNCIONALES: Descripción del servicio con atributos como calidad, radio funcional, etc.
3. DESCRIPCIÓN FUNCIONAL: La descripción declarativa, con los siguientes elementos:
 - a) Entradas
 - b) Salidas
 - c) Precondiciones
 - d) Efectos

Este modelo de descripción se usa tanto como para publicar un servicio como para solicitarlo. En el caso de necesitar un servicio, el cliente compila una descripción de un servicio ficticio que satisface sus necesidades, y hace una búsqueda de éste servicio usando el matching engine. De este modo, encontrará el más adecuado a sus necesidades.

4. Matching Engine

La visión de los autores de la WWW en el futuro es una amplia infraestructura de servicios Web soportada por un conjunto de registros que sirven de directorios. Estos registros guardan los anuncios de servicios que se activan, y soportan la búsqueda por un conjunto de funcionalidades. Este matching engine funcionará con un algoritmo que provea de las siguientes condiciones:

1. Soporte para flexible matching: Debe poder proveer de el servicio más adecuado a lo pedido usando match semántico (con ontologías).
2. Minimizar falsos positivos y falsos negativos, y además proveer al usuario alguna forma de parametrizar que tan riguroso quiere el match.

3. Debe incentivar tanto a clientes como a proveedores a ser precisos con sus anuncios/pedidos.
4. Debe ser eficiente. Cero gracia en estar esperando horas para un match.

Un match se logra cuando el servicio anunciado le sirve de algo al cliente que lo solicita. Esto sucede cuando hay coincidencia entre las entradas y salidas de la solicitud y el anuncio. El match de entradas asegura que el cliente le puede entregar al servicio los datos necesarios para operar, y el match de salidas asegura que las necesidades de información del cliente son satisfechas.

El algoritmo utilizado compara el pedido contra todos los anuncios del registro, cada match es evaluado para darle un puntaje, y se retorna la lista de calces ordenada.

Para hacer este calce, es necesario calzar todas las salidas de la petición contra las del anuncio. Si alguna de ellas falla, no hay calce. Para las salidas, es necesario que todas las entradas del anuncio calcen con las del pedido. Nuevamente, si falta alguna no hay calce. Pueden existir servicios que entreguen más salidas que las que se requieren, o pedidos que entreguen más entradas que las que necesita un servicio.

El *grado* del match depende de la relación entre los conceptos (tomada de ontologías) que se están comparando, y generalmente se reduce a la mínima distancia entre ellos en el árbol taxonómico. Esto nos define cuatro grados de match, ordenados por preferencia:

1. exacto: la salida del pedido (OutR) es equivalente a la salida del anuncio (outA).
2. plugin: outA subsume a outR.
3. subsume: outR subsume a outA.
4. nulo: no hay relación de subsunción entre los conceptos.

Lo más importante para la clasificación es el match de salidas. El match de entradas se usa más que nada para resolver empates de grado de salidas entre distintos servicios.

El algoritmo presentado permite hacer el match flexible pedido, basándose en inferencias sobre ontologías disponibles. Sin embargo, esta flexibilidad no impide que se descarten los anuncios que no sirven, y que acepte, pero con baja calificación, los anuncios que apenas cumplen. Además, se puede fijar en nivel mínimo aceptable de match (flexibilidad controlada por el usuario). Sobre la eficiencia, todavía no se termina el testeado de implementaciones.

5. Aplicación a UDDI

UDDI provee de un registro general de servicios web en internet. Los proveedores registran sus contactos y los servicios que prestan. Actualmente, UDDI soporta búsquedas por keywords o por tipo de servicios (ordenados mediante

una taxonomía de NAICS). Estas búsquedas no permiten hacer ningún tipo de inferencia o match flexible. La solución propuesta es proveer a UDDI del matching engine descrito para generar una capa semántica sobre los registros UDDI. De este modo, se puede buscar por keywords o capacidades. Al registrar un servicio sobre UDDI, se envía además su anuncio descrito semánticamente, de modo que se guarden tanto su descripción UDDI como su descripción declarativa. De este modo, es posible buscarlo en los registros UDDI (por keyword) o por match de capacidades. Esto le da un gran valor agregado a la búsqueda de servicios.

6. Discusión/Conclusión

El paper presenta un framework y algoritmo para generar un sistema de búsqueda de servicios por capacidades. Esta descripción usando DAML-S abre las puertas a la automatización de servicios. Que los servicios sean capaces de encontrarse entre sí provee de la base para que los servicios interactúen en forma automática, con mínima o nula intervención humana. Esta capacidad de los servicios de interactuar *sin necesidad de condificación ad-hoc* es otro paso más en dirección a una web semántica.